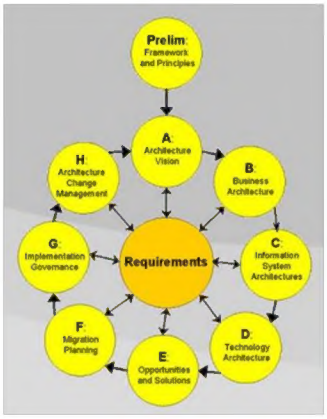

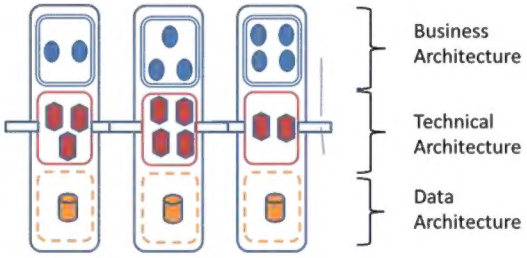

A Comparison of the Top Four Enterprise Architecture Approaches in 2014

by Roger Sessions and John deVadoss

October , 2014

Table of Contents

About the Authors 1

Introduction 2

Three Fundamental Questions 4

A Closer Look at the Four Approaches 4

Definition of Terminology 7

A Brief History of Enterprise Architecture 8

Case Study 12

The Perspective Centric Approach (Zachman) 15

The Process Centric Approach (TOGAF) 19

The Standardization Centric Approach (FEA) 26

The Capability Centric Approach (VRF/SIP) 33

Comparison 43

Conclusion 47

Glossary 49

References 52

2014 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

About the Authors

Roger Sessions (roger@objectwatch.com) is the world's leading expert in IT Complexity Analytics and the developer of The Snowman Architecture, a capability focused approach to enterprise architecture. He has written seven books, dozens of influential articles, and given lectures at conferences throughout the world. He has been a guest professor of IT Complexity Analytics at the University of the Andes and a two time keynote speaker at the highly prestigious Gartner Research Board on the topic of IT Complexity Management. His ground breaking work in Enterprise Architecture and IT has earned him recognition as a Fellow of the International Association of Software Architects.

Roger Sessions offers workshops and mentoring in Enterprise Architecture and the Snowman Practice. His website is www.objectwatch.com, and his blog is simplearchitectures.blogspot.com.

John deVadoss has over 17 years of experience in the software industry; he has been at Microsoft for over 13 years, all of it in the enterprise space – as a consultant, as a program manager in the distributed applications platform division, as an architect working with some of Microsoft's key partners, director of architecture strategy, leading technical strategy for the application platform, and most recently as product unit manager of patterns & practices. He is currently General Manager for the Enterprise Services Strategy program.

His areas of interest are broadly in distributed application architectures, data and metadata, systems management and currently on edge architectures (both services and access) aka cloud computing, but most of all in creating business value from technology investments. He has published books on topics such as Service-Oriented Architecture, Collaboration, and Cloud Computing among others. John has a MS in Computer Science from the University of Massachusetts at Amherst where he dabbled in AI, getting his degree in Machine Learning. He spent a year in the PhD program in Computer Science – and he still harbors visions of going back to complete the program at some point in his career.

Introduction

The year 2014 marks a full quarter century of enterprise architecture. In that time a number of enterprise architectural approaches have come and gone.

Should you care about a field that is twenty-five years old? It depends. This field was inaugurated to address two major problems in information technology that were twenty-five years ago already becoming apparent. The first problem was managing the increasing complexity of information technology systems. The second problem was the increasing difficulty in delivering real business value with those systems.

As you can imagine, these problems are related. The more complex a system, the less likely it is that it will deliver its promised business value. As you better manage complexity, you improve your chances of delivering real business value.

Today, four general approaches or understandings of enterprise architecture dominate the field. The *perspective centric approach* focuses on understanding different perspectives within the enterprise. The *process centric approach* focuses on the process that is used to define the deliverables. The *standardization centric approach* focuses on defining and enforcing the use of standard approaches throughout the enterprise. The *capability centric approach* focuses on understanding the capabilities of the organization (what it does) and which of those capabilities can be made more efficient through better use of technology or by outsourcing.

These four approaches share little except the name *enterprise architecture*. The goal of this white paper is to help you understand the differences between these four approaches so that you can make the best possible choice as to which will work best in your enterprise solving the problems that you need solved.

So should you care about this field? It depends on how you feel about positively impacting your organization's bottom line. If managing system complexity and delivering business value are key priorities for you, then you should care about enterprise architecture. If you are focused on maintaining or rebuilding IT's credibility in your organization or if you strive to promote the use of IT to maintain a competitive position in your industry, then you should continue reading this white paper. If these issues don't concern you, then these methodologies have little to offer.

As systems become more complex they generally require more planning. It is easy to see this in buildings. When Henry David Thoreau built his little cabin on Walden's Pond (shown in Figure 1) he embraced simplicity and needed no architect. If you are building New York City (shown in Figure 2) simplicity is out of the question and you will need architects.

Figure 1. Thoreau's cabin at Walden Pond, as drawn by Thoreau's sister, Sophia Thoreau

Figure 2. New York City

The relationship between complexity and planning for buildings and cities is similar for information systems. If you are building a simple, single-user, non-distributed system, you might not need any architects at all. If you are building an enterprise-wide, mission critical, highly distributed system, you might need a database architect, a solutions architect, an infrastructure architect, a business architect and an enterprise architect.

This paper is about the methodologies needed to develop the overall architectural vision for an organization. This is the responsibility of the *enterprise architect*. This is the architect who specializes in the broadest possible view of architecture within the enterprise. This is the

architect's architect, the architect who is responsible for coordinating the work of all of the other architects. Do you need such an architect? It all depends on what you are building: Thoreau's cabin or New York City.

Building a large, complex, enterprise-wide information system without an enterprise architect is like trying to build a city without a city planner. Can you build a city without a city planner? Probably. Would you want to live in such a city? Probably not.

Of course, hiring a city planner does not guarantee you will build a livable city, it merely improves your chances. Similarly, having an enterprise architect does not guarantee you will realize the business benefits from enterprise architecture. There are many examples of failed enterprise architectures in the world today and most of them had enterprise architects (probably dozens!) Architectural methodologies can help, but they only go so far. We will also discuss some of the reasons for these failures, and how to avoid them.

Three Fundamental Questions

Before any enterprise embarks on a program of enterprise architecture, there are three fundamental questions the enterprise must be able to answer. The questions are these:

- How do we define enterprise architecture?
- What will be the business benefit of doing enterprise architecture?
- How we will measure the business benefit of having done enterprise architecture?

If an enterprise cannot answer these three questions, then it has no justification for starting (or continuing) an enterprise architecture program. Unfortunately, as we will soon see, it is easier to ask these questions that it is to answer them.

A Closer Look at the Four Approaches

Before we explore the dominant approaches to Enterprise Architecture we need to start by defining what we mean by Enterprise Architecture. It turns out that this term is not well defined.

In September 2012, I (Roger) started a thread on the Linked-In group The Enterprise Architecture Network asking practicing Enterprise Architects to define Enterprise Architecture. Over the next six months more than 150 replies were posted. While the range of the definitions made it clear that there is no common agreement, four approaches to enterprise architecture seem to emerge[1].

[1] Each of these authors has kindly given us permission to include their quotes. However this does not imply that they would agree with this four-way classification of enterprise architecture or that they would agree with our characterization of their quotation.

The first approach is illustrated by this comment:

> EA, as the name states, is the architecture that is the integrated blueprint of the enterprise. EA, simply put, describes the enterprise from many angles that serve its various stakeholders.
>
> - Adrian Grigoriu

We call this approach *perspective centric*, because it is focusing on the need to ensure that all of the different perspectives within the organization have been adequately considered. The methodology that best represents this approach is The Zachman Framework for Enterprise Architectures™ usually just called Zachman.

The second approach is illustrated by this comment:

> EA is a rigorous model of the motivations, structures, information, processes, and systems of an enterprise created for the purpose of decision support.
>
> - Nick Malik[2]

We call this approach *process centric* because it implies the need to focus on the process used to model the enterprise. The methodology that best represents this approach today is The Open Group Architecture Framework (TOGAF™).

The third approach is illustrated by this comment:

> Enterprise Architecture is
>
> a) a plan for the fundamental structure of systems which comprise and support the enterprise together with the context of those systems and the principles governing the development and evolution of those systems
>
> b) the activity of creating the architecture described above through a process of analyzing, assessing and documenting the context & environment of the systems, researching options and making decisions for the systems components, structures & principles; together with the communication of that architecture to interested parties and governance activities to ensure the architecture decisions are adhered to across the organization.
>
> - Richard Hillier

---

[2] This is expanded upon in his blog at http://blogs.msdn.com/b/nickmalik/archive/2010/08/08/a-reasonable-canonical-definition-of-enterprise-architecture.aspx

We call this approach *standardization centric*, because it is focusing on the need to define standard practices, reference models, and common services that are understood through the enterprise. The methodology that best represents this perspective is Federal Enterprise Architecture (FEA).

The fourth approach is illustrated by this comment:

> EA is about optimizing technology investment to best deliver desired business capabilities.
>
> - Ed Klein

We call this approach *capability centric*, because it focuses on the capabilities within the enterprise and how best to improve them with technology. The methodology that best represents this perspective is the Value Realization Framework (VRF) coupled with Simple Iterative Partitions (SIP).

As you can see, these four approaches are quite different. Which should your organization follow? It depends on what you are trying to accomplish. A good starting point to making the choice is to look at how each of these four perspectives might answer the three fundamental questions and see which most closely resonates with your enterprise. Figure 3 summarizes the answers.

EA Approach	What is EA?	What is the Business Benefit of EA?	How will the Business Benefit of EA be measured?	Representative Technology
Perspective Centric	A description of each important issue from each important perspective.	EA shows how issues at one perspective map to issues at another perspective.	Better alignment of the different perspectives.	Zachman
Process Centric	A process that is followed to build business or IT systems.	The EA process is the same across industries.	Improvements in IT architectures.	TOGAF
Standardization Centric	A set of enterprise standards.	Standardization fosters reuse.	More reuse.	FEA
Capability Centric	A roadmap for improving capabilities.	EA pinpoints opportunities to use technology more effectively.	More competitive business capabilities.	VRF/SIP

Figure 3. Four Approaches on the Three Fundamental Questions

Definition of Terminology

Before we get too far into comparing the approaches that make up the enterprise architect's toolkit, we need to define some terms. This is especially important in an article that is comparing Enterprise Architectural approaches since the different approaches sometimes use similar terms to mean different things.

For example, we have two approaches that describe themselves as enterprise architectural frameworks: the Zachman Framework for Enterprise Architectures and The Open Group Architectural Framework (TOGAF). Yet these two methodologies share little in common other than the words *enterprise*, *architecture*, and *framework*.

So we will start by defining the terms as used in this white paper. Those definitions marked with asterisk are taken mostly from IEEE-1471-2000 [01], whose definitions are used where they exist and make sense.

architect – One whose responsibility is the design of an architecture and the creation of an architectural description.

architectural artifact – A specific document, report, analysis, model, or other tangible that contributes to an architectural description.

architectural description* – A collection of products (artifacts) that taken together document an architecture.

architectural framework – A skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other, and provides generic definitions for what those artifacts might look like.

architectural methodology – A generic term that can describe any structured approach to solving some or all of the problems related to architecture.

architectural process – A defined series of actions directed to the goal of producing either an architecture or an architectural description.

architectural taxonomy – A methodology for organizing and categorizing architectural artifacts.

architecture* – The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

enterprise architecture – An architecture in which the system in question is the whole enterprise, especially the business processes, technologies, and information systems of the enterprise.

Now that we have a common understanding of these key terms, we can take you through the history of enterprise architecture methodologies, discuss the problems these methodologies are trying to solve, and compare the top four methodologies in terms of their approach and their relationship to each other.

A Brief History of Enterprise Architecture

The field of enterprise architecture started in 1987 with the publication in the IBM Systems Journal of an article titled, "A framework for information systems architecture," by J.A. Zachman. In that paper, Zachman laid out both the challenge and the vision of enterprise architectures that would guide the field for the next 25 years. The challenge was to manage the complexity of increasingly distributed systems. As Zachman said:

> The cost involved and the success of the business depending increasingly on its information systems require a disciplined approach to the management of those systems. [02]

Zachman's vision was that business value and agility could best be realized by a holistic approach to systems architecture that explicitly looked at every important issue from every important perspective. His multi-perspective approach to architecting systems is what Zachman originally described as an *information systems architectural framework* and soon renamed to be an *enterprise architecture framework*.

Zachman was a major influence on one of the earliest attempts by a branch of the U.S. Government, the Department of Defense, to create an enterprise architecture. This attempt was known as the Technical Architecture Framework for Information Management (TAFIM) [03]. TAFIM was introduced it in 1994.

The promise of enterprise architectures, such as TAFIM, to better align technical projects with business need was noticed by no less a body than the U.S. Congress. Most likely influenced by the promised benefits of TAFIM, Congress in 1996 passed a bill known as the Clinger-Cohen Act [04], also known as the Information Technology Management Reform Act, which mandated that all federal agencies take steps to improve the effectiveness of their IT investments. A CIO Council, consisting of CIOs from all major governmental bodies, was created to oversee this effort.

In April 1998, the CIO Council began work on its first major project, the Federal Enterprise Architecture Framework (FEAF). Version 1.1 [05] of this framework was released in September, 1999. This document contained some innovative ideas such as "segmented architectures", that is, architectural focus on segmented subsets of the larger enterprise.

Over time responsibility for federal enterprise architecture moved from the CIO Council to the Office of Management and Budget (OMB). In 2002 the OMB evolved and renamed the FEAF methodology as the Federal Enterprise Architecture (FEA). We will describe FEA in greater detail in the section on standardization focused approaches.

Two documents that have come out in the last few years that are relevant to the future of FEA.

The first document is the *25 Point Implementation Plan to Reform Federal Information Technology Management* by then United States Chief Information Officer, Vivek Kundra [06]. This document was released in December 2010. Although Kundra is no longer in this position, this document still wields considerable influence. The 25 Point Plan (as it is often called) is Kundra's vision for how IT can be revamped in the Federal government. It is worth noting that enterprise architecture in general and FEA specifically is never mentioned. This makes it seem that at least at the CIO level of the government, EA is not considered strategically important.

Another issue that will likely surface in the next few years is the 25 Point Plan's emphasis on cloud architecture. Kundra saw the cloud as a strategically critical move for the Federal Government. The current version of FEA appears to be pretty poorly aligned with the cloud platform [07].

The second relevant document is the *Common Approach to Federal Enterprise Architecture* released in May 2012 [08]. Given that this is one of the few documents issued by the executive branch on the topic of FEA in recent years, one would expect a robust update of a major framework. Instead, the Common Approach is an anemic document (barely 40 pages without appendices) that spends more time justifying FEA than showing how it is used.

Reading between the lines of these two documents, it seems that the energy behind FEA is dissipating. We expect that over the next few years we will see a major shift in the attitude toward FEA within the Federal Government.

Despite the very significant enterprise architectural activity in the Federal Government (one could argue that no organization has spent more money attempting to develop an enterprise architecture than the U.S. Government), progress has been slow and success stories are overshadowed by higher profile failures. In 2004, a full eight years after the Clinger-Cohen act mandated the use of effective IT planning processes, the General Accounting Office (GAO) reported the following:

> Only 20 of 96 agencies examined had established at least the foundation for effective architecture management. Further, while 22 agencies increased in maturity since 2001, 24 agencies decreased in maturity and 47 agencies remained the same. [09]

Since January, 2005, the GAO has severely chastised a number of U.S. Agencies for failures in their adoption or use of enterprise architecture. A few examples include the FBI [10], the Department of Defense [11], the Department of Homeland Security [12], and NASA [13].

In 1998, four years after TAFIM (remember TAFIM?) was introduced and two years after it became codified as Clinger-Cohen, TAFIM was officially retired by the Department of Defense.

The work done on TAFIM was turned over to The Open Group. They morphed it into a new standard that is today known as The Open Group Architectural Framework, better known by its acronym, TOGAF. TOGAF has gone through a number of iterations. The current version of TOGAF and the one we will be discussing is 9.1. We will discuss TOGAF in the section on the *process centric approach*.

About the time that Zachman was developing his original ideas on Enterprise Frameworks, another thread was starting up that eventually would be highly influential in enterprise architecture. This thread was known as *object-oriented programming*.

Object-oriented (OO) programming is a particular style of writing computer programs such that packages of code appear analogous to business objects; you ask an object to perform an action, pass the object the information it needs to do so, and then get information back that is the result of the action having been performed. For example, if one is implementing a purchase requisition system, one might have a Purchase_Requisition object that you can ask to Process_Requisition, passing information in the form of parameters about the requisition, and receiving information back indicating the success or failure of the requisition.

The reason object-oriented programming is interesting in the context of enterprise architecture is that this style introduced the concept that software systems could be analogous to business processes. You could, for example, have a software object, Process_Requisition, that mirrored the business process. There was the hint of a correspondence between the business capability and the supporting technology.

In theory, OO systems offered a separation between what an object did (its behavior) and how it did it (its implementation.) The separation was adjudicated by the object's interface. In practice, this separation was poorly understood and rarely respected.

OO systems became even more interesting in the mid-1990s when objects became distributed. With distribution, they took on a new name, *components*. It now became possible to ask a component to do something for you even if that component lived in a different address space, a different machine, or even in a different enterprise. This required that the idea of the interface, the definition of what the component does, be taken much more seriously. Components were more and more described as black boxes. You put well defined things in and got well defined things out. What happened inside was opaque.

By 1996 there were two major competing component (distributed object) systems. On the Microsoft side, there was DCOM. On the other side, there was CORBA (Common Object Request Broker Architecture.) CORBA was owned by a consortium known as the Object Management Group (OMG). OMG was largely spearheaded by IBM, Sun, and Oracle with Microsoft conspicuously absent. In 1996, DCOM (Microsoft) and CORBA (OMG) were two distinct component worlds that could not communicate with each other.

By 2002 the component world was starting to adopt a vendor-independent format for sending messages between components. This format was known as Simple Object Access Protocol (SOAP.) Now for the first time it was possible for both CORBA and DCOM components to work together. A new term was coined to describe this approach: Service-Oriented Architectures (SOA.) What had originally been distributed objects and then components were now *services*.

Over the next five years there was increasing interest in merging the business processes with the technology supporting them. Business processes became more and more described in service-like language and services became more and more described in business process-like language. A new term was born: capabilities. A capability was something the business did that provided value and included the underlying supporting technology.

In 2003, for example, when my (Roger's) book *Software Fortresses* [14] came out, I described a software fortress as consisting of strong boundaries which included the business processes, technical support systems, and underlying data. The idea of a software fortress was an early definition of what today we know as a capability.

Microsoft was also very interested in the evolution of components into capabilities. They of course were early pioneers in components through their DCOM technology. They were further influenced by a 2008 paper co-authored by one of their consultants, Ric Merrifield, in the Harvard Business Review titled *The Next Revolution in Productivity* [15]. This article introduced a number of ideas that influenced the capability focused approach to enterprise architecture, most importantly, the idea that a business should be described in packages (capabilities) of what it does, not how it does it.

As the distinction between business and technology became increasingly blurred, it was clear that this new way of thinking belonged in a discipline embracing both business and technology. The obvious landing place was enterprise architecture. By 2010 all of the important ideas of the capability centric enterprise architecture were in place and we can say that a new approach to enterprise architecture was born.

This brings us up to the present. As you can see, the field of enterprise architecture is a still evolving field. Figure 4 summarizes this history with an enterprise architecture timeline.

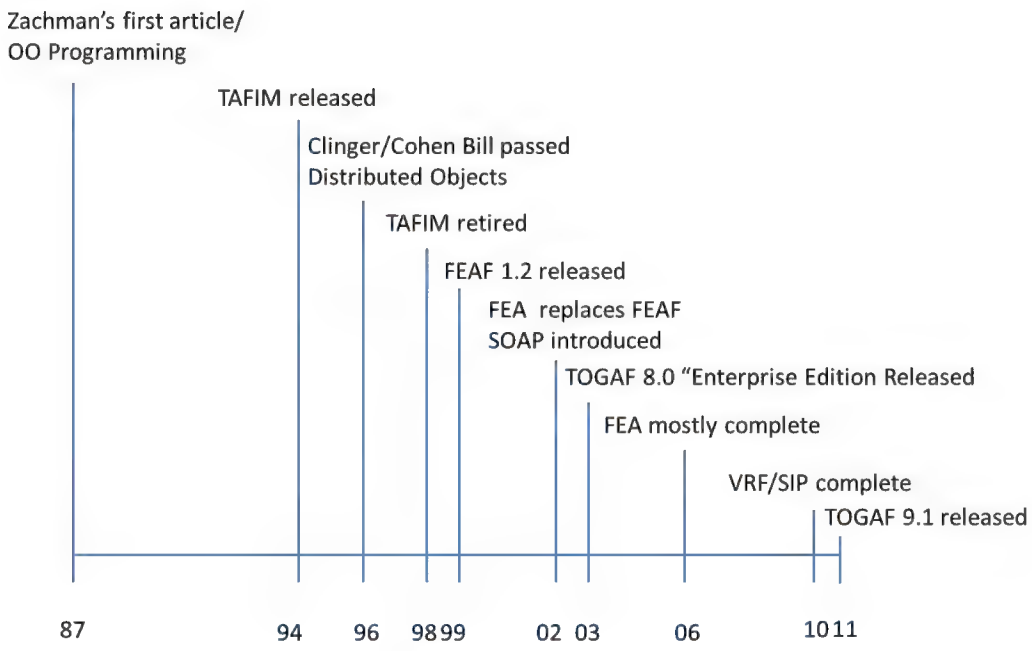

Figure 4. Enterprise Architecture Timeline

Now let's look more closely at today's main enterprise architectural approaches and introduce a case study that will be used in this white paper.

Case Study

So that we can compare and contrast the four major approaches to enterprise architectures, we are going to illustrate how each would approach a similar scenario. This fictitious scenario is a composite of several enterprises with which we have worked over the past several years. So while it is fictitious, it is very realistic. We'll first describe the scenario.

MedAMore is a chain of drug stores. It started as a regional chain in 1965. In 2000, it developed an innovative software system that enabled it to run drug stores very efficiently. It called this system MedAManage, or MAM. MAM incorporated some innovate business ideas such as patient relationship management, inventory management, automated insurance billing, and even utility optimization.

MAM consisted of three programs: MAM/Store, which ran on a small computer at a drug store, MAM/Warehouse, which ran on a server in a regional warehouse, and MAM/Home, which ran on a large server at the home office.

These three programs communicated through files that were transferred from one location (e.g., a store) to another location (e.g., a regional warehouse). When reliable communications

lines existed, file transfers could occur through FTP. The system was also flexible enough to accommodate transfers through courier, where necessary.

By 2005, MedAMore was doing quite well, in part because of the cost cutting moves enabled by the MAM system. MedAMore decided to begin expansion. To do this, it purchased three regional chains. With these purchases, MedAMore extended its reach through the southeast quadrant of the US.

By 2007, it was clear that the same software systems that had initially fueled MedAMore's success were now hampering its future. Some of the problems MedAMore were running into were the following:

- MAM/Store required regional specializations. For example, different insurance plans needed to be supported in different regions, and these all required changes to the MAM/Store module.
- The regional warehouses that had been acquired through acquisition each had different ways of receiving orders from the retail stores and different procedures from ordering supplies from the wholesalers. Each of these differences required changes to the MAM/Warehouse module.
- The file transfer approach to information sharing that had worked so well when MedAMore consisted of 30 drugstores, one regional warehouse, and one home office, were turning out to be difficult to coordinate among 200 drugstores, four regional warehouses, two geographic offices, and one home office. Files were often delivered late, sometimes not at all, and occasionally multiple times. This made it difficult for the home office to access reliable, up-to-date financial information especially in the area of sales and inventory.

It was clear to MedAMore management that the MAM system needed many enhancements. However upgrading this system was difficult. Each of the three modules (store, warehouse, and home office) was huge, inefficient and cumbersome as they included functionality for everything that each entity might need.

The modules had grown to over one million lines of code each. It was difficult to change one function without impacting others. All of the functions accessed a single database and changes to one record definition could ripple through the system in an unpredictable fashion. Changing even a single line of code required a rebuild of the entire multi-million line module.

MedAManage had become MedANightmare. Debugging was difficult. Software builds were torturous. Installing new systems was hugely disruptive.

These technical problems soon created internal conflicts within the home office of MedAMore. The business side of MedAMore wanted to acquire two more regional chains, but IT was still struggling to bring the existing acquisitions on-line.

This resulted in a rapidly growing divide between the business and the technical sides of MedAMore. The business side saw IT as reducing business agility. The technical side saw the business side as making impossible demands and blamed it for refusing to consult IT before entering into acquisition discussions.

The distrust had reached such a point that by 2010, the CIO was no longer considered part of the executive team of MedAMore. The business side distrusted IT and tried to circumvent it at every opportunity. The technical side built their IT systems with little input from the business folks. Several large and expensive IT initiatives were ignored by the business side and were eventually abandoned.

By 2011, MedAMore was in crisis. It clearly needed to revamp its technical systems to make them easier to specialize for regional requirements. This was going to be an expensive proposition and MedAMore couldn't afford for the effort to fail.

Just as importantly, MedAMore also had to rebuild its internal relationships. The constant bickering and distrust between business and IT was impacting morale, efficiency, and profitability. A company that only five years earlier was an industry leader in profitability in large part because of its innovative use of IT was now struggling to stay out of the red in large part because of the inflexibility of those same IT systems.

Cath, the CEO of MedAMore, desperately needed a solution. At a CEO conference she heard how many of her peers were using enterprise architectures to build stronger partnerships between their technical and business groups and deliver more cost effective IT systems that enabled business agility.

Cath decided that this approach merited further investigation. She asked Irma, her CIO, to prepare a recommendation on the use of enterprise architecture within MedAMore. Irma was impressed with the approach, but recognized that any such initiative needed to be driven from the top and needed to involve the business side from the start.

On Irma's recommendation, Cath called a meeting with Bret, the Vice-President of Business, and Irma. Cath announced that she had decided to create a common enterprise architecture for MedAMore that would unite its technical and business people. This common enterprise architecture would be named MedAMore-Enterprise Architecture, or MAM-EA. Once completed, MAM-EA would drive all new IT investment and ensure that every dollar invested in IT was delivering the maximum value to the business.

Cath knew that MAM-EA was a bet-the-company decision for MedAMore. The MAM-EA vision *had* to work. Cath was depending on Bret (the business side) and Irma (the IT side) to make it work.

So that is the problem. Now let's see how each of the EA approaches might provide a solution for MedAMore.

The Perspective Centric Approach (Zachman)

Those that follow a perspective centric approach to enterprise architecture are focusing mainly on different perspectives within the enterprise and making sure that the different perspectives lineup. The best representative of this genre is none other than the venerable Zachman framework.

The first thing we need to understand about the Zachman Framework is that is isn't a framework, at least by our definition of a framework. According to the American Heritage Dictionary, a framework is defined as:

> A structure for supporting or enclosing something else, especially a skeletal support used as the basis for something being constructed; An external work platform; a scaffold; A fundamental structure, as for a written work; A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality. [16].

A *taxonomy*, on the other hand, is defined as:

> The classification of organisms in an ordered system that indicates natural relationships; The science, laws, or principles of classification; systematics; Division into ordered groups or categories. [17]

The Zachman "Framework" is actually a taxonomy for organizing architectural artifacts (i.e., design documents, specifications, models) that takes into account both who the artifact targets (e.g., business owner, builder) and what particular issue (e.g., data, functionality) is being addressed.

As John Zachman retrospectively described his work:

> The [Enterprise Architecture] Framework as it applies to Enterprises is simply a logical structure for classifying and organizing the descriptive representations of an Enterprise that are significant to the management of the Enterprise as well as to the development of the Enterprise's systems. [18]

Many proponents of the Zachman Framework see it as cross disciplinary, with influence extending far beyond IT. One popular book on Zachman, for example, says:

> …in due course, you will discover that the Framework exists in everything you do, not only IT projects. When you thoroughly understand the Framework, you can become more effective in everything you do. This means *everything*. This statement is not made lightly. [19]

John Zachman himself told me in an interview that I (Roger) conducted with him:

> ...the Framework schema has been around for thousands of years and I am sure it will be around for a few more thousands of years. What changes is our understanding of it and how to use it for Enterprise engineering and manufacturing. [20]

Zachman originally explained his IT taxonomy using the building industry as an analogy. In that industry, architectural artifacts are implicitly organized using a two dimensional organization. One dimension is the various "players in the game." For a physical building, some of these players are the owner (who is paying for the project), the builder (who is coordinating the overall construction), and a zoning board (who is ensuring that construction follows local building regulations).

A building architect prepares different artifacts for each of these players. Every player demands complete information, but what constitutes completeness differs for the different players. The *owner* is interested in a complete description of the functionality and aesthetics of the building. The *builder* is interested in a complete description of the materials and construction process. The owner doesn't care about the placement of studs in the walls. The builder doesn't care how the bedroom windows line up with the morning sun.

As Zachman said in his original article:

> ... each of the architectural representations differs from the others in *essence*, not merely in level of detail. [02]

The second dimension for architectural artifact organization is the descriptive focus of the artifact, the what, how, where, who, when, and why of the project. This dimension is independent of the first. Both the builder and the owner need to know *what*, but the owner's need to know *what* is different than the builder's need to know *what*. What *what* is what depends on who is asking the question.

In his first paper and Zachman's subsequent elaboration in 1992 [22], Zachman proposed that there are six descriptive foci (data, function, network, people, time, and motivation) and six player perspectives (planner, owner, designer, builder, subcontractor, and enterprise). From the business owner's perspective "data" means business entities. This can include information about the entities themselves, such as customers and products, or information about relationships between those entities, such as demographic groups and inventories. If you are talking to a business owner about data, this is the language you should use.

From the perspective of the person implementing the database, "data" does not mean business entities, but rows and columns organized into tables and linked together by mathematical joins and projections. If you are talking to a database designer about data, don't talk about customer demographic groups, talk about third-normal relational tables.

It's not that one of these perspectives is better than the other or more detailed than the other or of a higher priority than the other. *Both* of these perspectives on data are critical to a holistic understanding of the system's architecture. As Zachman said:

> We are having difficulties communicating with one another about information systems architecture, because a *set* of architectural representations exists, instead of a single architecture. One is not right and another wrong. The architectures are different. They are additive and complementary. There are reasons for electing to expend the resources for developing each architectural representation. And there are risks associated with *not* developing any one of the architectural representations.[02]

We discussed the historical importance of the Zachman Framework in the history section. Here we will discuss the actual framework itself and how it could be used to help build MAM-EA, the problem proposed in the case study section.

As mentioned earlier, the Zachman Framework consists of six functional foci, each considered from the perspective of a major player. The Zachman framework underwent a major facelift in 2011 and most people are familiar with the old style Zachman framework. The Zachman framework as it is portrayed today is shown in Figure 5.

As you can see from Figure 5, there are 36 intersecting cells in a Zachman grid, one for each meeting point between a player's perspective (e.g., architect) and a descriptive focus (e.g., process definition). As we move horizontally (e.g., left to right) in the grid, we see different descriptions of the system, all from the same player's perspective. As we move vertically in the grid (e.g., top to bottom), we see a single focus, but change the player from whose perspective we are viewing that focus.

There are three suggestions of the Zachman grid that can help MedAMore in the development of MAM-EA.

The first suggestion of the Zachman taxonomy is that every architectural artifact should live in one and only one cell. There should be no ambiguity about where a particular artifact lives. If it is not clear in which cell a particular artifact lives, then there is most likely a problem with the artifact itself.

As MedAMore begins accumulating artifacts in the development of MAM-EA, it can use the Zachman grid to clarify the focus of each of these artifacts. For example, artifacts relating to a service-oriented architecture live mostly in the third row (architect's perspective). They generally will not be of interest to the business owner (Bret, in the MedAMore case study).

The second suggestion of the Zachman taxonomy is that an architecture can be considered a *complete* architecture only when every cell in that architecture is complete. A cell is complete when it contains sufficient artifacts to fully define the system for one specific player looking at one specific descriptive focus.

When every cell is populated with appropriate artifacts, there is a sufficient amount of detail to fully describe the system from the perspective of every player looking at the system from every possible angle (descriptive focus). So MedAMore can use the Zachman grid to ensure that appropriate discussions are occurring between all of the important stakeholders of MAM-EA.

The third suggestion of the Zachman grid is that cells in columns should be related to each other. Consider, for example, the inventory column (the first column) of the Zachman grid. From business management's (Bret's) perspective, *inventory* is information about the business. From the engineering perspective, *inventory* is rows and columns in the database.

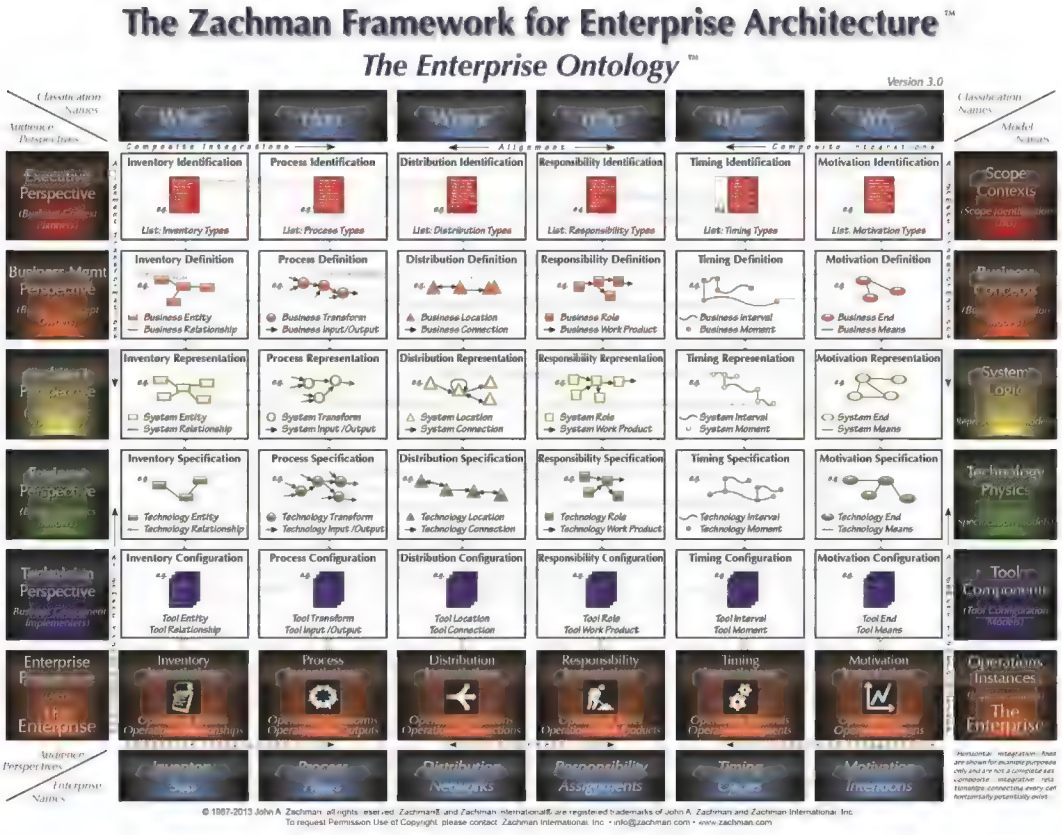

Figure 5. Zachman Grid[3]

While the business owner thinks about inventory quite differently than the engineer, there should be some relationship between these perspectives. Somebody should be able to follow

[3] This diagram is copyright by John A. Zachman and Zachman International®. Published with the permission of John A. Zachman and Zachman International®, Inc. - www.zachman.com

Bret's business requirements and show that the database design is, in fact, being driven by those requirements. If Bret has requirements that are not traceable down to the database design, then we must ask if the business needs will be met by this architecture. On the other hand, it there are database design elements that do not trace back to business requirements, then we might ask if we have included unnecessary design at the database level.

So we can see five ways that the Zachman grid can help in the development of MAM-EA. It can help:

1. ensure that every stakeholder's perspective has been considered for every descriptive focal point.
2. improve the MAM-EA artifacts themselves by sharpening each of their focus points to one particular concern for one particular audience.
3. ensure that all of Bret's business requirements can be traced down to some technical implementation.
4. convince Bret that Irma's technical team isn't planning on building a bunch of useless functionality.
5. convince Irma that the business folks are including her IT folks in their planning.

But Zachman by itself is not a complete solution for MedAMore. There are far too many issues that will be critical to MAM-EA's success that Zachman does not address. Zachman does not give us a step-by-step process for creating a new architecture. Zachman doesn't even give us much help in deciding if the future architecture we are creating is the best architecture possible. For that matter, Zachman doesn't even give us an approach to show a need for a future architecture. For these, and other issues, we are going to need to look at other methodologies.

The Process Centric Approach (TOGAF)

Proponents of the process centric approach to enterprise architecture believe that a good enterprise architecture is dependent on a good process. It is critical therefore that we bring together the world's best knowledge in defining what this process should look like. The best known representative of this genre is The Open Group Architecture Framework which is best known by its acronym, TOGAF.

TOGAF is, as you might have guessed from its name, owned by The Open Group [23]. The most recent version of TOGAF is 9.1 released in 2011. TOGAF's view of an enterprise architecture is shown in Figure 6.

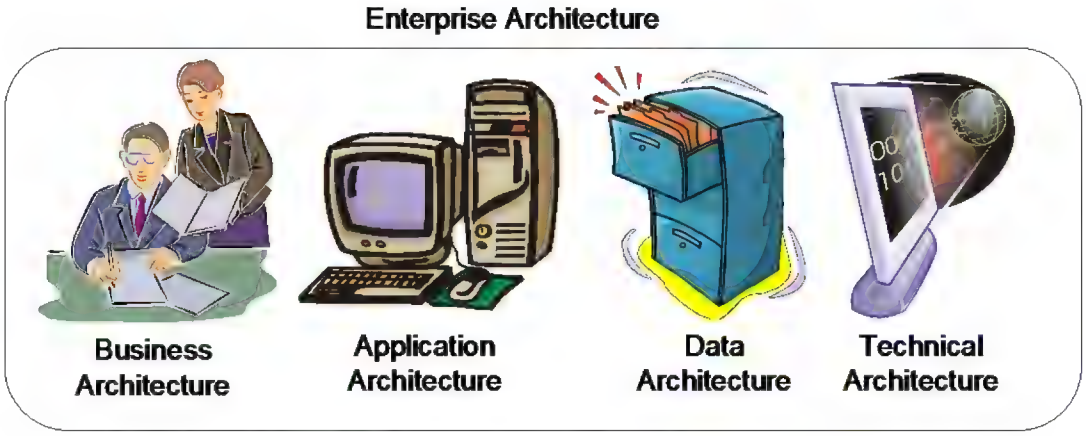

Figure 6. TOGAF's Enterprise Architecture

As shown in this figure, TOGAF divides an enterprise architecture into four categories, as follows:

- Business architecture — Describes the processes the business uses to meet its goals
- Application architecture — Describes how specific applications are designed and how they interact with each other
- Data architecture — Describes how the enterprise datastores are organized and accessed
- Technical architecture — Describes the hardware and software infrastructure that supports applications and their interactions

TOGAF describes itself as a "framework", but the most important part of TOGAF is the Architecture Development Method, better known as ADM. ADM is a recipe for creating an architecture. A recipe can be categorized as a *process*. Given that ADM is the most visible part of TOGAF, we categorize TOGAF overall as an *architectural process* rather than either an *architectural framework*, as The Open Group describes TOGAF, or a methodology, as it describes ADM.

Viewed as an architectural *process*, TOGAF complements Zachman, which, recall, was categorized as an architectural *taxonomy*. Zachman tells you how to categorize your artifacts. TOGAF gives you a process for creating them.

TOGAF views the world of enterprise architecture as a continuum of architectures, ranging from highly generic to highly specific. It calls this continuum the Enterprise Continuum. It views the process of creating a specific enterprise architecture, such as MAM-EA, as moving from the generic to the specific. TOGAF's ADM provides a process for driving this movement from the generic to the specific.

TOGAF calls most generic architectures *Foundation Architectures*. These are architectural principles that can, theoretically, be used by any IT organization in the universe.

TOGAF calls the next level of specificity *Common Systems Architectures*. These are principles that one would expect to see in many — but perhaps not all — types of enterprises.

TOGAF calls the next level of specificity *Industry Architectures*. These are principles that are specific across many enterprises that are part of the same domain — such as, in our MedAMore case study, all pharmaceutical enterprises.

TOGAF calls the most specific level the *Organizational Architectures*. These are the architectures that are specific to a given enterprise, such as MedAMore.

Figure 7 shows the relationship between the Enterprise Continuum and the Architectural Development Methodology (ADM).

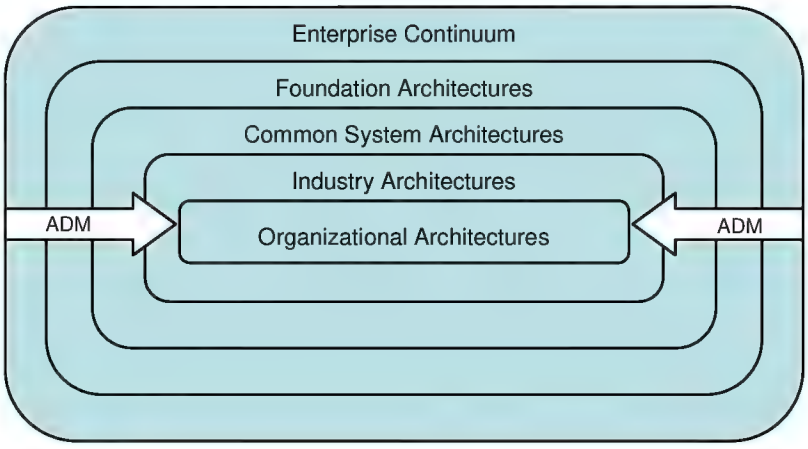

Figure 7. The TOGAF Enterprise Continuum

TOGAF defines the various knowledge-bases that live in the Foundation Architecture. Two that you might run into are the *Technical Reference Model* (TRM) and the *Standards Information Base* (SIB). The TRM is a suggested description of a generic IT architecture. The SIB is a collection of standards and pseudo-standards that The Open Group recommends that you consider in building an IT architecture.

TOGAF presents both the TRM and the SIB as suggestions; neither is required. In our view, both the TRM and the SIB are flawed for the same reason: they are biased toward application *portability* at the expense of application *interoperability* and application *autonomy*. We consider this an outdated view of technical architectures.

For an organization such as MedAMore, TOGAF largely boils down to the Architecture Development Method (ADM). Individuals within MedAMore will be exposed to the

Enterprise Continuum, the SIB, and the TRM (as well as a few other TOGAF features) which is why we discussed them. But the day-to-day experience of creating an enterprise architecture will be driven by the ADM, a high level view of which is shown in Figure 8.

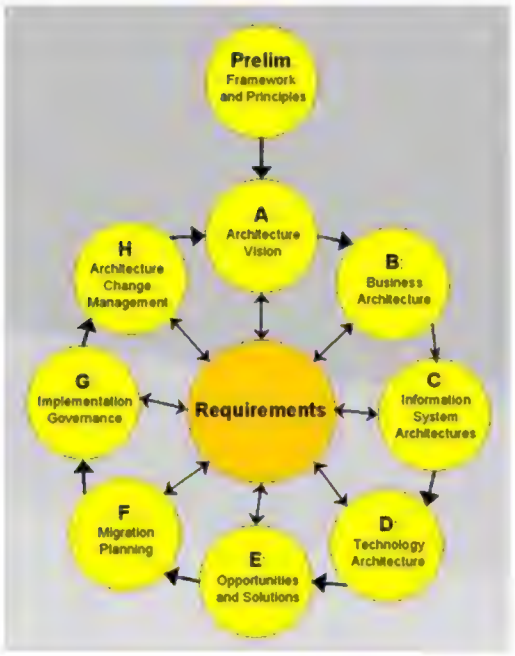

Figure 8. The TOGAF Architecture Development Method (ADM)As shown in Figure 8, the TOGAF ADM consists of eight phases that are cycled through after an initial "priming of the pump." We'll take you through these phases as they could be applied to the MedAMore case study. But, before MedAMore can start the ADM, it needs to gain some experience with TOGAF. MedAMore will have two choices on how it can get this experience.

First, MedAMore can train itself in TOGAF. MedAMore can download the TOGAF documentation [24] which describes all of TOGAF including the ADM in considerable detail. It can purchase books on TOGAF [25]. There is probably more free and inexpensive information available about TOGAF than about all other architectural methodologies combined.

Second, MedAMore can buy expertise in TOGAF. There are consultants that specialize on TOGAF and that have earned Open Group certification [26]. Since MedAMore wants to minimize any chances of failure, it has chosen to call in a TOGAF consultant. MedAMore has brought in Teri, an Open Group certified TOGAF architect. Remember the other players at MedAMore are Cath, the CEO of MedAMore, Bret, the Business VP, and Irma, the CIO.

In the Preliminary Phase, Teri meets with the major players at MedAMore to introduce the TOGAF process. Her three goals in the preliminary phase are to:

1. make sure everybody is comfortable with the process,

2. modify the TOGAF process as necessary to fit within the MedAMore culture, and
3. set up the governance system that will oversee future architectural work at MedAMore.

Teri will work closely with Brett to understand the business philosophy, business models, and strategic drivers of MedAMore. She will work closely with Irma to define the architectural principles that drive technological architectures at MedAMore and document those principles using the TOGAF-recommended format.

In some organizations, achieving buy-in on the need for enterprise architecture could be very difficult. This is especially true when the effort is driven from the IT organization, and even more so when there is a history of discord between the business and the technical sides of the organization. MedAMore has such a history of animosity. However, it has another fact going for it from which Teri should take heart: the effort is not driven by IT, but is driven by Cath, the CEO. This gives the project high visibility and creates a positive incentive for cooperation from all sides.

As soon as Teri and MedAMore have completed the Preliminary Phase, they are ready to start Phase A. Phase A begins, at least in theory, with a *Request for Architecture Work* from some organization within MedAMore. This document includes the business reasons for the request, budget and personnel information, and any constraints that need to be considered. Because MedAMore has never done a *Request for Architecture Work*, Teri will probably need to work with the sponsoring organization in creating such a request.

As soon as the Request for Architecture Work (or some equivalent) has been received, Teri (the TOGAF consultant) starts MedAMore on Phase A. In Phase A, Teri will ensure that the project has the necessary support within MedAMore, define the scope of the project, identify constraints, document the business requirements, and establish high level definitions for both the baseline (starting) architecture and target (desired) architecture.

These baseline and target definitions will include high-level definitions on all four of the EA sub-architectures shown back in Figure 5 — namely, business, technology, data, and application architectures.

The culmination of Phase A will be a Statement of Architecture Work which must be approved by the various stakeholders before the next phase of the ADM begins. The output of this phase is to create an architectural vision for the first pass through the ADM cycle. Teri will guide MedAMore into choosing the project, validating the project against the architectural principles established in the Preliminary Phase, and ensure that the appropriate stakeholders have been identified and their issues have been addressed.

The Architectural Vision created in Phase A will be the main input into Phase B. Teri's goal in Phase B is to create a detailed baseline and target business architecture and perform a full

analysis of the gaps between them. She will work primarily with Bret (or Bret's team) to achieve this.

Phase B is quite involved — involving business modeling, highly detailed business analysis, and technical requirements documentation. A successful Phase B requires input from many stakeholders. The major outputs will be a detailed description of the baseline and target business objectives, and gap descriptions of the business architecture.

Phase C does for the information systems architecture what Phase B does for the business architecture. In this phase, Teri works primarily with Irma (or her team). TOGAF defines nine specific steps, each with multiple sub-steps:

1. Develop baseline data architecture description
2. Review and validate principles, reference models, viewpoints, and tools
3. Create architecture models, including logical data models, data management process models, and relationship models that map business functions to CRUD (Create, Read, Update, Delete) data operations
4. Select data-architecture building blocks
5. Conduct formal checkpoint reviews of the architecture model and building blocks with stakeholders
6. Review qualitative criteria (e.g., performance, reliability, security, integrity)
7. Complete data architecture
8. Conduct checkpoint/impact analysis
9. Perform gap analysis

The most important deliverable from this phase will be the Target Information and Applications Architecture.

Phase D completes the technical architecture — the infrastructure necessary to support the proposed new architecture. This phase is completed mostly by engaging with Irma's technical team.

Phase E evaluates the various implementation possibilities, identifies the major implementation projects that might be undertaken, and evaluates the business opportunity associated with each. The TOGAF standard recommends that Teri's first pass at Phase E "focus on projects that will deliver short-term pay-offs and so create an impetus for proceeding with longer-term projects."

This is good advice in any architectural methodology. Therefore, Teri should be looking for projects that can be completed as cheaply as possible while delivering the highest perceived value. A good starting place to look for such projects is the organizational pain-points that initially convinced Cath (the MedAMore CEO) to adopt an enterprise architectural-based strategy in the first place. These pain-points, described earlier, included difficulties in completing regional/warehouse specialization and unreliability in data sharing.

Phase F is closely related to Phase E. In this Phase, Teri works with MedAMore's governance body to sort the projects identified in Phase E into priority order that include not only the cost and benefits (identified in Phase E) but also the risk factors.

In Phase G, Teri takes the prioritized list of projects and creates architectural specifications for the implementation projects. These specifications will include acceptance criteria and lists of risks and issues.

The final phase is H. In this Phase, Teri modifies the architectural change management process with any new artifacts created in this last iteration and with new information that becomes available.

Teri is then ready to start the cycle again. One of the goals from the first cycle should be information transfer, so that Teri's services are required less and less as more and more iterations of the cycle are completed.

Much of the results of the TOGAF process will be determined as much by the Teri/MedAMore relationship as it will by the TOGAF specification itself. TOGAF is meant to be highly adaptable, and details for the various architectural artifacts is sparse. As one book on TOGAF says:

> TOGAF is not wholly specific with respect to generated documents; in fact, it provides very little in the way of prescriptive document templates – merely guidelines for inputs and outputs. [27]

The TOGAF specification is also flexible with respect to the phases. As the specification itself says:

> One of the tasks before applying the ADM is to review its components for applicability, and then tailor them as appropriate to the circumstances of the individual enterprise. This activity might well produce an "enterprise-specific" ADM. [28]

TOGAF allows phases to be done incompletely, skipped, combined, reordered, or reshaped to fit the needs of the situation. So it should be no surprise if two different TOGAF certified consultants end up using two very different processes, even when working with the same organization.

TOGAF is even more flexible about the actual generated architecture. In fact, TOGAF is, to a surprising degree, "architecture-agnostic". The final architecture might be good, bad, or indifferent. TOGAF merely describes *how* to generate an enterprise architecture, not necessarily how to generate a *good* enterprise architecture. For this, you are dependent on the experience of your staff and/or TOGAF consultant. People adopting TOGAF hoping to

acquire a magic bullet will be sorely disappointed (as they will be with any of the methodologies).

The Standardization Centric Approach (FEA)

Proponents of the standardization centric approach to enterprise architecture believe that the main purpose of enterprise architecture is to define a common language that is used throughout the enterprise. This common language will foster cooperation and cooperation will foster reuse, the ultimate goal of this approach. The best known representative of this genre is the Federal Enterprise Architecture (FEA.)

The Federal Enterprise Architecture (FEA) is the latest attempt by the federal government to unite its myriad of agencies and functions under a single common and ubiquitous enterprise architecture. FEA is still in its infancy as most of the major pieces have only been available since 2006. However, as discussed in the history section, it has a long tradition behind it, and, if nothing else, has many failures from which it has hopefully learned some valuable lessons.

FEA is the most complete of all the methodologies discussed so far. It has both a comprehensive taxonomy, like Zachman, and an architectural process, like TOGAF. FEA can be viewed as either a methodology for creating an enterprise architecture or the result of applying that process to a particular enterprise — namely the U.S. Government. We will be looking at FEA from the methodology perspective. Our particular interest here is how can we apply the FEA methodology to private enterprises.

Most writers describe FEA as simply consisting of five reference models, one each for performance: business, service, components, technical, and data. It is true that FEA has these five references models, but there is much more to FEA than just the reference models. A full treatment of FEA needs to include all of the following:

- A perspective on how enterprise architectures should be viewed (the *segment* model, that we will describe shortly)
- A set of reference models for describing different perspectives of the enterprise architecture (the five models, mentioned above)
- A process for creating an enterprise architecture
- A transitional process for migrating from a pre-EA to a post-EA paradigm
- A taxonomy for cataloging assets that fall in the purview of the enterprise architecture
- An approach to measuring the success of using the enterprise architecture to drive business value

You can see that the FEA is about much more than models. It includes everything necessary to build an enterprise architecture for probably the most complex organization on earth: The U.S. Government. As the FEA-Program Management Office (FEAPMO) says, FEA, taken *in toto*, provides:

> ... a common language and framework to describe and analyze IT investments, enhance collaboration and ultimately transform the Federal government into a citizen-centered, results-oriented, and market-based organization as set forth in the President's Management Agenda. [29]

While it might be a stretch to imagine that anything short of divine intervention could "transform the Federal government into a citizen-centered, results-oriented, and market-based organization", there is at least hope that some of the FEA methodology could help our beleaguered MedAMore corporation deal with its much more mundane problems. So, let's take a look at what FEA has to offer.

The FEA Perspective on EA

The FEA Perspective on EA is that an enterprise is built of *segments*, an idea first introduced by FEAF [30]. A segment is a major line-of-business functionality, such as human resources. There are two types of segments: *core mission area segments* and *business services segments*.

A *core mission area segment* is one that is central to the mission or purpose of a particular political boundary within the enterprise. For example, in the Health and Human Services (HHS) agency of the federal government, *health* is a core mission area segment.

A *business services segment* is one that is foundational to most, if not all, political organizations. For example, *financial management* is a business services segment that is required by all federal agencies.

Another type of enterprise architecture asset is an *enterprise service*. An enterprise service is a well-defined function that spans political boundaries. An example of an enterprise service is *security management*. Security management is a service that works in a unified manner across the whole swath of the enterprise.

The difference between *enterprise services* and *segments*, especially *business service segments*, is confusing. Both are shared across the entire enterprise. The difference is that business service segments have a scope that encompass only a single political organization. Enterprise services have a scope that encompass the entire enterprise.

In the federal government, for example, both HHS and the Environmental Protection Agency (EPA) use the business service segment *human resources*. However the people that are managed by human resources is a different group for HHS than it is for the EPA.

Both HHS and the EPA also use the enterprise service *security management*. But the security credentials that are managed by the security management service are not specific to either of those agencies. Security credentials are only managed effectively when they are managed at the scope of the enterprise.

Resist the temptation to equate either *segments* or *services* with services as in *service oriented architectures*. There are two reasons such a comparison would be flawed. Firstly, enterprise services, business-service segments, and core mission-area segments are all much broader in focus than services found in service oriented architectures. Secondly, segments are an organizational unit for an *enterprise architecture*, whereas services are an organizational unit for *technical implementations*. As organizational units for an enterprise architecture, their depth includes not just the technical, but also the business and the data architectures.

One final note about segments: although segments *function* at the political (that is, agency) level, they are *defined* at the enterprise (that is, government) level. Enterprise services, of course, both function and are defined at the enterprise level.

The fact that segments are defined globally facilitates their reuse across political boundaries. One can map out the usage of segments across the political boundaries of the enterprise and then use that map to seek opportunities for architectural reuse. Figure 9, for example, shows a segment map of the federal government from the FEA Practice Guide [31]. As you can see, there are many segments (the vertical columns) that are used in multiple agencies and any or all of these are good candidates for sharing.

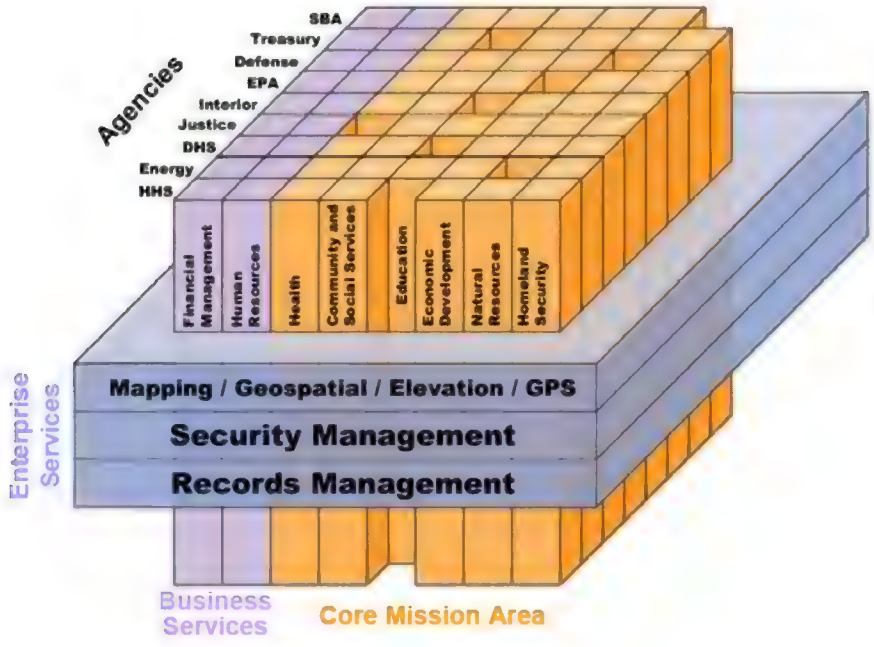

Figure 9. Segment Map of the Federal Government

FEA Reference Models

The five FEA reference models are all about establishing common languages. The goal here is to facilitate communication, cooperation, and collaboration across political boundaries. According to the FEAPMO:

> The FEA consists of a set of interrelated "reference models" designed to facilitate cross-agency analysis and the identification of duplicative investments, gaps and opportunities for collaboration within and across agencies. Collectively, the reference models comprise a framework for describing important elements of the FEA in a common and consistent way. [32]

Why do we need a common language? Consider this exchange:

James: Do you have a torch I can borrow?
Roger: No, I'm afraid not.
James: Do you know where I can get one?
Roger: The hardware store in town should have one.

So James goes out to the hardware store and buys himself a torch. He returns.

Roger: Did you get your torch?
James: Yes, here it is.
Roger: That's not a torch! That's a flashlight. Why didn't you say so? I have one you could have borrowed.
James: Well why didn't *you* say so?

The problem, of course, is that James comes from England where what we call a *flashlight*, they call a *torch*. And when we hear *torch*, we think of a *blowtorch*. Although we both speak English, we don't necessarily speak the same English. The result is that James goes out and unnecessarily spends money on something that we could have lent him.

This is exactly the problem that the FEA Reference Models are trying to solve on a much larger scale. Suppose the Internal Revenue Service (IRS) decides it needs a *demographics* system to track taxpayer data. They ask around to see if anybody has one they can modify for their purposes. Nobody does.

Little do they know that right next door the Government Printing Office (GPO) has a perfectly good demographics system that is almost exactly what the IRS needs. They just happen to call it a *customer analytics* system.

So, the IRS goes out and builds its system from scratch rather than simply modifying the one already built (and paid for) by the GPO. And, in doing so, the IRS will waste considerably more money than James spent on his unnecessary flashlight.

This, in a nutshell, is the goal of the five FEA reference models: to give standard terms and definitions for the domains of enterprise architecture and thereby facilitate collaboration and sharing across the federal government. The five reference models are as follows:

1. The *Business Reference Model (BRM)* gives a business view of the various functions of the federal government. For example, the BRM defines a standard business capability called *water resource management* that is a sub-function of *natural resources* that is considered a *line-of-business* of the broader business area *services for citizens*. [32]
2. The *Components Reference Model (CRM)* gives a more IT view of systems that can support business functionality. For example, the CRM defines a *customer analytics* system that we described in the hypothetical interchange between the Internal Revenue Service and the Government Printing Office above. [32]
3. The *Technical Reference Model (TRM)* defines the various technologies and standards that can be used in building IT systems. For example, the TRM defines *HTTP* as a *protocol* that is a subset of a *service transport* that is a subset of *service access and delivery*. [32]
4. The *Data Reference Model (DRM)* defines standard ways of describing data. For example, the DRM defines an *entity* as something that *contains attributes* and *participates-in relationships*. [33]
5. The *Performance Reference Model (PRM)* defines standard ways of describing the value delivered by enterprise architectures. For example, the PRM describes *quality* as a technology measurement area that is defined as "the extent to which technology satisfies functionality or capability requirements." [34]

FEA Process

The FEA Process is primarily focused on creating a segment architecture for a subset of the overall enterprise (in FEA's case, the enterprise is the federal government and the subset is a governmental agency) and is described in the FEA Practice Guidance [35]. We discussed the FEA vision on enterprise segments earlier. The overall segment-architecture development process is (at a very high level) as follows:

- Step 1: Architectural Analysis – Define a simple and concise vision for the segment and relate it back to the organizational plan.
- Step 2: Architectural Definition – Define the desired architectural state of the segment, document the performance goals, consider design alternatives and develop an enterprise architecture for the segment including business, data, services, and technology architectures.
- Step 3: Investment and Funding Strategy – Consider how the project will be funded.
- Step 4: Program Management Plan and Execute Projects – Create a plan for managing and executing the project, including milestones and performance measures that will asses project success.

FEA Success Measurement

The FEA framework for measuring organizational success in using enterprise architecture is defined in the Federal Enterprise Architecture Program EA Assessment Framework 2.1 [36]. Federal agencies are rated as to their overall maturity levels in three main categories:

1. architectural completion — maturity level of the architecture itself
2. architectural use — how effectively the agency uses its architecture to drive decision-making and
3. architectural results — the benefits being realized by the use of the architecture

Category: Architectural Completion
Description: This category measures the architectural maturity of an enterprise's architecture in terms of performance, business, data, service, and technology. This includes an assessment of the architectural artifacts and both the baseline (existing) and target (goal) architectures.

Level	Name	Description
1	Initial	The enterprise is using informal and ad-hoc EA processes. Some architectural artifacts for a given architectural level may exist, but the levels are not linked, or the linkage is incomplete.
2	Baseline	The enterprise has developed a baseline (as-is) architecture. The architecture has enterprise-wide scope and the linkages between levels are well established and clearly articulated.
3	Target	The enterprise has developed both a baseline architecture (as described above) and a target (goal) architecture. The target architecture is aligned to enterprise-wide goals and organizational responsibilities. The target architecture addresses the priorities and performance objectives identified in the enterprise business plan.
4	Integrated	The enterprise has developed at least vertically partitioned architecture that has been approved by the business owner in writing. The relevant organization(s) within the enterprise are actively migrating toward the relevant architecture.
5	Optimized	The enterprise has developed multiple vertically partitioned architectures that support core mission business functions, all approved by the appropriate business owners.

Figure 10. OMB Ranking of Architectural Completion,
Adapted for Private Sector by Roger Sessions

OMB assigns each agency a success rating, based on its scores in each category and a cumulative score, as follows:

- Green — The agency rates quite well in the *completion* area (it has a quite mature enterprise architecture). It also rates well in both the *use* area (it is effectively using

that enterprise architecture to drive ongoing strategy) and the *results* area (the usage of that architecture is driving business value).

- Yellow — The agency rates quite well in the *completion* area. It also rates well in either the *use* area or the *results* area.
- Red — The agency either does not have a completed architecture and/or is not effectively using that architecture.

The framework is interesting beyond the confines of the public sector. The category ratings can be fruitfully adapted by many enterprises to assess the maturity level of their own architectural efforts. Figure 9, for example, shows our own interpretation of the OMB maturity rankings for *architectural completion* as adapted for the private sector. Similar adaptations can be created for *architectural usage* and *architectural results*.

FEA Applied to MedAMore

Now that we have taken you through the FEA approach, let's see what this might mean to MedAMore. Let's assume that Cath (MedAMore's CEO) has heard about FEA and how it is promising to streamline the federal government. If it can do this for the federal government, she reasons, surely it can do this for her company.

Cath hires a consultant, Fred, who is an expert on FEA. Fred's job is to show MedAMore how to do FEA — of course, not the real FEA, but FEA as it might be applied to the private sector. Cath introduces Fred to Bret (the business VP) and Irma (the CIO) and tells them to build her a MEA — FEA adopted for MedAMore.

Keep in mind that Cath has taken quite a risk. No other company to date has attempted to apply FEA to the private sector, and even the experience of using FEA within the public sector is nominal at best.

The first thing that Fred will want to do is build enthusiasm for MEA. Keep in mind that he is coming into an organization in which the business folks barely speak to IT folks. If MEA is going to succeed, it needs to transform not only processes, but people. He will want to create a series of seminars explaining the value of the soon-to-be-defined MEA and how MEA will benefit not only MedAMore as a whole, but the individual units specifically.

Fred will next build a governance structure, MedAMore's equivalent to FEAPMO. We'll call this group MEAPMO. MEAPMO will own MEA, including the processes, the models, and the architecture itself.

The next thing that Fred will likely do is to create reference models that can be used by all of the organizations across MedAMore. The five reference models from FEA can serve as a starting point. Some, such as the Technical Reference Model, might be usable with few modifications. Others, such as the Business Reference Model, will require extensive

renovation. He shouldn't do these in excruciating detail, but create starting points and build them up as MEA evolves.

Next, Fred will probably want to create a description of the segment architecture as it applies to MedAMore. Note that he will not be doing a complete segment architecture — just a high level description. The actual process of completing the architecture will be a constantly evolving project.

By this point, a lot of work will have been done with few results. Fred will probably want to take a first pass at a segment architecture process. FEA's process will be a good starting point, but will require specialization to MedAMore at the detail level (such as who the team members are and what form the generated artifacts should take).

Now Fred will test drive the process with the first segment delivery. He will need to build a team and then lead this team in analyzing and prioritizing the segments — mapping them to business value, determining their architectural options, delivering the work, and, perhaps most important, measuring the results. These measurements will be critical in building momentum for future work.

Soon after completing the first segment, Fred might decide that it is time to measure the progress of the different groups in MedAMore in using MEA effectively. To do so, Fred needs a yardstick to measure the success of the different groups within MedAMore in driving business value with MEA. Fred thus leads MEAPMO in building a MedAMore equivalent to the Federal Enterprise Architecture Program EA Assessment Framework [36]. This yardstick will be Cath's main tool for ensuring that the different groups are taking MEA seriously and that her investment is paying off.

And, finally, once Fred has completed this process, he will start the process again. Each iteration will result in new segments being delivered, more business value being generated, and more substance being added to the MEA methodology. At least, this is the theory. As we said earlier, with MEA, we are working at the bleeding edge.

The Capability Centric Approach (VRF/SIP)

So far we have looked at three approaches to enterprise architecture: perspective centric, process centric, and standardization centric.

The *perspective centric approach* (e.g. Zachman) focuses on understanding the different perspectives of the different stakeholders and how these perspectives are related to each other. If we can just understand how the business perspective maps to the IT perspective, then we will ensure that the business needs are met by the IT systems. So it is critical that we ensure that every significant issue is considered from every possible perspective.

The *process centric approach* (e.g. TOGAF) focuses on the methodology used. If we can just fine tune the method we use for mapping business systems to IT systems, then we will build better IT systems. So in this approach, it is considered critical that we use standardized methods for mapping that we then tune and retune.

The *standardization centric approach* (e.g. FEA) focuses on describing the enterprise is a commonly understood language and then using that language to describe how all the pieces fit together. If we can just describe how all of the business processes are related to each other and how these map to IT systems, then we will build better IT systems. So it is critical that we create and then maintain these detailed mappings using a common language.

Now we look at the fourth approach to enterprise architecture, the *capability centric approach*. In capability centric enterprise architecture, the enterprise is modeled as a collection of capabilities. Each capability performs some important function for the enterprise. The goal of enterprise architecture in this approach is to identify which capabilities are in need of improvement and then devise a plan for improving them.

To understand this better, let's look more closely at a generic capability shown in Figure 11.

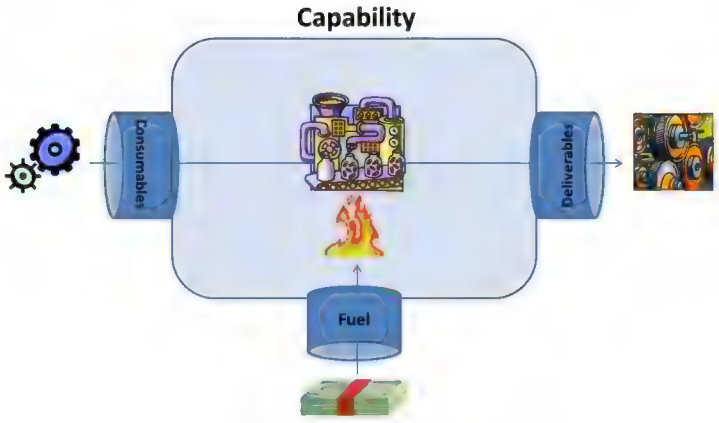

Figure 11. Generic Capability

As shown in Figure 11, a capability consumes resources (*consumables*) and creates resources (*deliverables*). It expends fuel (money, assets) to transform consumables into deliverables. Very few enterprises consist of a single capability. Usually there is a network of capabilities with the deliverables of one capability becoming the consumables of other capabilities. This view of an enterprise is shown in Figure 12.

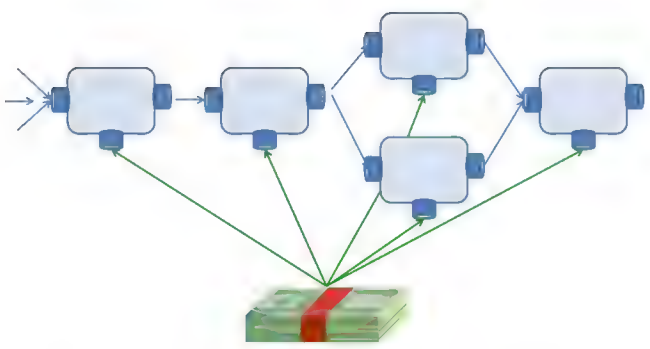

Figure 12. The Capability View of an Enterprise

We can also see this as a value chain[4] of capabilities and it relates to a high level process view of the organization. Standard process hierarchies such as APQC[5] come with industry standard benchmarks that help us understand which capabilities are performing well or below an industry average or competitor.

In enterprises of any significant size, capabilities are hierarchical. This means capabilities are built out of lower level capabilities, as shown in Figure 13.

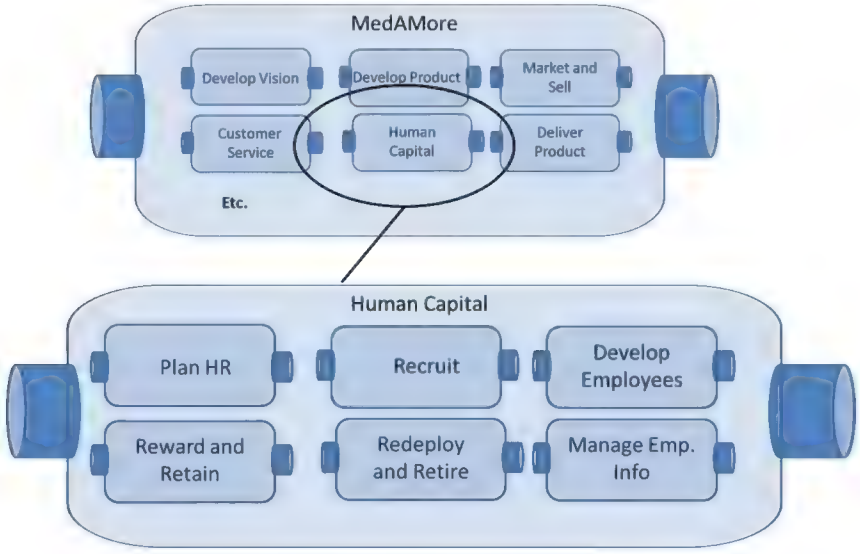

Figure 13. Hierarchy of Capabilities

[4] After Porter et al.

[5] APQC (American Productivity & Quality Center) is a member-based nonprofit and one of the world's leading proponents of business benchmarking, best practices, and knowledge management research.

Any capability is expected to generate value. The value of a capability is given by the following formula:

Value (capability) = Value (deliverables) – Value (consumables) – Value (fuel)

In the capability focused approach to enterprise architecture, our goal is to realize the maximize value from the capability.

Deciding which capabilities to change or improve is central to this approach so that the "fuel" is used in the most effective way as characterized by the desired outcomes. We agree on a set of "value dimensions" with which we can compare the range of possible improvements or changes proposed. Not only can we show how the change supports a larger set of goals (business change, business objectives, responding to business drivers) but we can bind the contribution of the change to the business objectives and articulate a value story.[6]

Let's consider an example. One of the capabilities of human resources is *recruitment*. The input consumables for recruitment are *interested candidates* which includes every person who is interested in working for the enterprise. The output deliverables are *qualified candidates*. This is the much smaller pool of people who are a good match for the open positions in the enterprise. So the *recruitment* capability transforms interested candidates into qualified candidates. The fuel for the transformation is money, the universal fuel of most capabilities. Money is needed to pay salaries, run the physical facilities, and acquire the technology that will aid the transformation.

If we look to APQC, there is a significant difference between top and bottom performers regarding the cost per new hire.[7] Although organizations want to manage their cost per new hire through effective sourcing and recruiting mechanisms, this metric must also be balanced against maintaining a high quality of employee and high retention rates among top performers. Let's assume that we have determined that the *recruitment* capability value is too low, perhaps costing too much to transform *interested candidates* into *qualified candidates*. There are generally two solutions to this problem. The first is to outsource the capability. The second is to make better use of technology to reduce the cost of the transformation. If *outsourcing* is the chosen path, the problem is turned over to another group (i.e. another capability) that specializes in negotiating outside contracts. If *making better use of technology*

---

[6] Tools such as the Cranfield School of Management Benefits Dependency Network (see "Managing the Realization of Business Benefits from IT Investments" by Professor Joe Peppard) or Strategy Maps (after Kaplan) illustrate this binding so that the change story can be related to clear business objectives.

[7] Review the Source, Recruit, and Select Employees Open Standards Benchmarking Assessment available at http://www.apqc.org/benchmarking-portal/osb/source-recruit-and-select-employees

is the chosen path, then the enterprise architect drives the use of technology to reduce the cost of the transformation.

So in the capability focused approach to enterprise architecture our goal is to realize the maximum value from the enterprise capabilities. This high level goal of *value realization* can be subdivided into these sub-goals:

- Identify the value chains in the organization with objectives, stakeholders and measures.
- Identify the capabilities of the organization:
    - Find high potential capabilities, that is, capabilities that can deliver much higher value than they are currently delivering (VRF Context).
    - Determine which high potential capabilities can be addressed with a technology component, which may be most if not all. (VRF Validate)
    - Prioritize the high potential capabilities based on overall needs of the enterprise and estimate value from the overall value chain aligned with stakeholder objectives. (VRF Prioritize)
- Confirm stakeholder support and commitment for the change.
- Plan a change project with business, technology, people, process and workplace perspectives and ensure the organization has the ability to execute the whole plan. (VRF Initiative Planning)

Then and only then:

- Plan a strategy t increase the value of those capabilities.
- Ensure the technology solution is closely aligned with the business needs.
- Ensure that the technology solution is the best possible technology solution.
- Deliver and implement the solution.
- In concert with the People, Process and Organizational change work streams, actively drive and manage the realization of value and objectively demonstrate that value to the business was realized.(VRF Value Realization)

Given this general background to the capability focused approach to enterprise architecture, let's see how we would approach MedAMore's problems using representative technologies. We'll describe the process using two techniques which fit into the component focused approach. The first is the Value Realization Framework (VRF), a framework developed by Microsoft. The second is Simple Iterative Partitions (SIP), a methodology developed by ObjectWatch.

There are three reasons we are using these independently developed approaches to illustrate the capability focused approach. First, each of these addresses different aspects of a full capability value realization exercise. Second, using two different techniques illustrates the common approach of using different methods for different phases of a problem solution. And

third, it turns out that the two authors of this white paper are particularly familiar with these two approaches. John leads the development of VRF and Roger is the developer of SIP.

We'll start with an overview of how Value Realization Framework (VRF) and Simple Iterative Partitions (SIP) work together and then we will walk through the MedAMore scenario. Figure 14 shows the relationship between VRF and SIP.

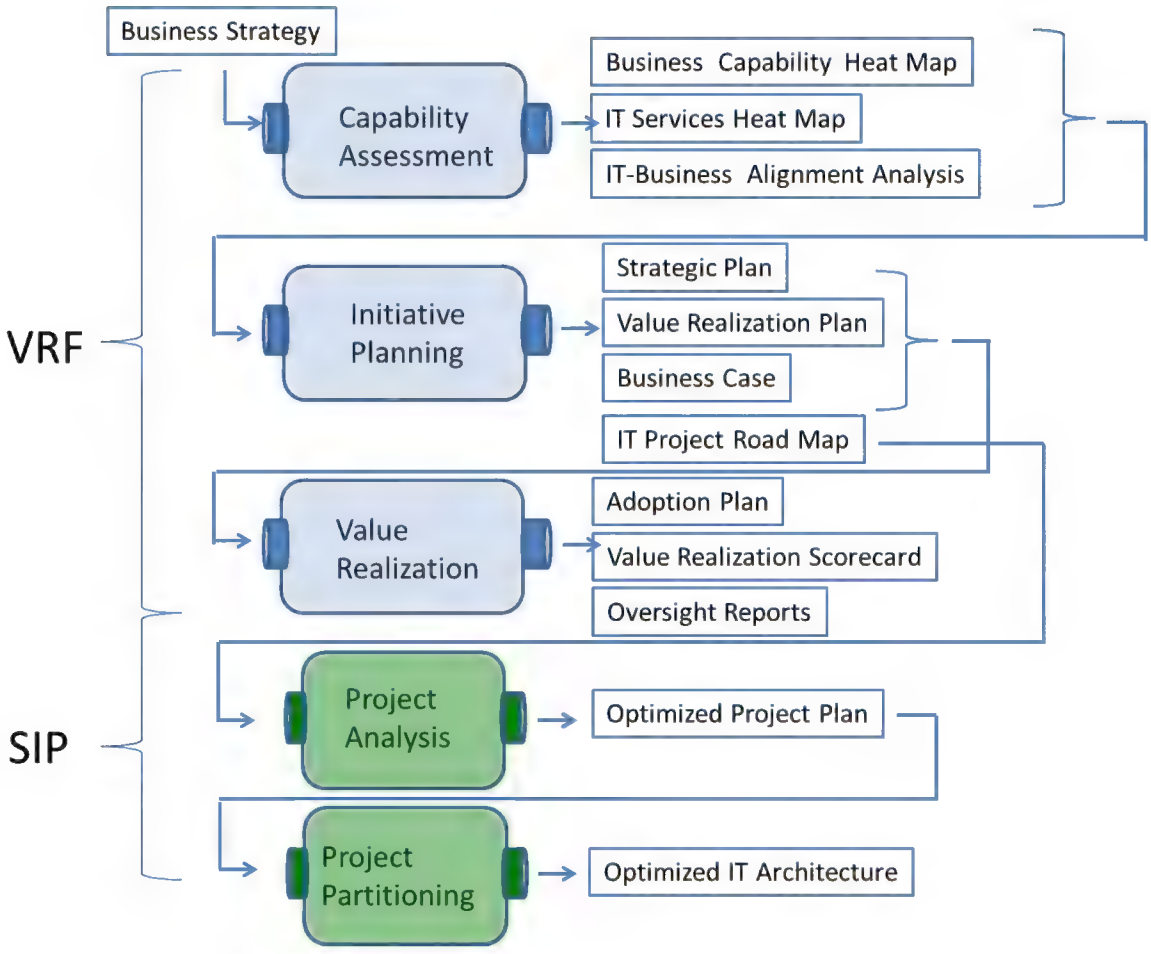

Figure 14. Relationship between Value Realization Framework (VRF) and Simple Iterative Partitions (SIP)

The intersection between VRF and SIP comes at the Initiative Road Map, one of the outputs of the VRF analysis. This road map is likely to include a number of recommended initiatives. When one or more of these recommended IT projects exceeds about one million dollars, then SIP comes into play. It turns out that one million dollars is a critical number. It is the point at which IT failure becomes a common problem.

Success rates for IT projects are quite high when the IT projects are one million dollars or less. Projects this size or smaller typically have at least a 75% success rate. Once the size reaches one million dollars, success rates drop off markedly. By the time the sizes reach ten million dollars, success rates drop to less than 10%[8].

SIP is specifically designed to address the issue of the high failure rates of large IT projects. It provides a methodology for finding the optimal way to partition and phase the large complex project into small simpler projects. Since each of the small simple projects is under the magic one million dollar number, success rates stay high.

Given this background, let's see how the MedAMore scenario plays out with VRF/SIP. In this scenario MedAMore brings in an outside VRF domain expert named Vic. Since it is clear that the resulting projects are going to be multi-million dollar projects, Vic recommends that Sally, a SIP specialist, also be part of the project.

Vic starts with the first major segment of VRF, Assessment. The starting point for the assessment phase is setting the context of the engagement. Vic will work with the MedAMore executive team to clearly define MedAMore's business strategy, define a high level capability map of the organization, and determine which of the identified capabilities will most benefit from a technological makeover. Once these capabilities have been identified, Vic works with MedAMore to define a collection of IT initiatives that seem to be good candidates for helping MedAMore better achieve its business strategy.

One advantage of the capability approach is that it gives a very fast way to identify those areas in the organizations that are in most need of a technology makeover while still maintaining a picture of how those areas are related to other areas in the organization. This allows Vic to collect just as much detail as he needs to help MedAMore drive value without getting bogged down in non-critical details.

Vic develops a capability map for MedAMore and, with the input of the executive team, identifies *Deliver Products and Services* as the capability in most need of help. Vic and MedAMore dive into this capability in more detail. The problems are localized to the lower level capability *Procure Materials and Services* and within that to the capability *Order Materials and Services.*

Once the capability (or capabilities) that most need attention have been identified, a number of candidate initiatives are explored. Vic and the MedAMore team focus on three:

- The process for collecting inventory information from the warehouse is old and fragile. It runs on large expensive computer systems and relies on FTP transmission of

---

[8] For more information on the relationship between project size and project success rates, see the presentation The Relationship Between IT Project Size and Failure Rates [37]

data files that have poorly documented formats that often break and require human adjudication.
- The process for placing orders with suppliers requires expensive code updates whenever a new supplier is added.
- The process for receiving materials at the loading dock requires humans to validate each item received against placed orders.

In a few short weeks, Vic and MedAMore have created a capability map of the enterprise and have highlighted the area of the organization where there is the best possible opportunity to increase productivity. Over the next three weeks, each of these candidate initiatives is explored in more depth. Some of the questions that are answered include the following:

- What is the current state of each of these initiatives?
- What is the desired future state of each of these initiatives?
- What would be the actual business benefit of each of these?
- How would this benefit be measured?
- What are the business challenges involved with achieving the future states?
- Which IT systems need to change to support these changes?

Vic and MedAMore now have enough information on each of the proposed initiatives to make an informed decision about how each should be prioritized. It is decided that the process for collecting information from the warehouses will be given the highest priority for these reasons:

- Twenty thousand dollars is lost each month due to inventory excesses. These excesses occur when inventory information is lost in transmission.
- The hardware platform that runs the collection function costs over one hundred thousand dollars per year in maintenance.
- The hardware platform is old and requires massive cooling facilities. The cost of the facilities to house the hardware platform costs ten thousand dollars per month.
- The software is difficult to maintain and requires a dedicated staff of three programmers and an operator at a combined salary of forty thousand dollars per month.

In addition to these direct costs of the antiquated software system, there are lost opportunity costs. MedAMore sees the opportunity to reduce its transportation costs by at least one hundred and fifty thousand dollars per month by splitting one of its larger warehouses into two smaller regional warehouses. This cannot be done with the current software system.

When Vic and MedAMore look at the proposed future system, they envision a cloud-based solution that will cost one thousand dollars per day and that can be maintained by a staff of less than one full time equivalent. The projected cost of the project is two million dollars.

The annual cost of the current system is thus $2.7 million, including the lost opportunity costs. The annual cost of the new system will be about $468 thousand, including both personnel and cloud rental. The annual savings will thus be $2.3 million. Given that the system will cost $2 million to build, the system will save MedAMore $11.5 million over five years for a return of investment of 5.75 to 1 and will pay for itself in less than a year. Besides the financial payoff, the new system will allow MedAMore to make some highly desirable business process changes (breaking up the large warehouse system.)

This is the logical approach that Vic and MedAMore use to assign this project such a high priority. None of their other candidate initiatives can pay off such a large dividend in such a short time. This is not to say these other candidate initiatives will not be done, just that they won't be done as the first priority.

So far, the only approach that has come into play is the Value Realization Framework (VRF.) But now we run into a problem. The anticipated cost of the new system is $2 million. We know from industry data that IT projects over $1 million in cost run a high risk of failure. What can we do to reduce the risk of failure?

This is where Sally, the SIP analyst comes in. Sally will work with the domain experts at MedAMore to analyze the business functions that have been identified during the application of VRF. She will then use the SIP algorithms to partition the large $2 million dollar system into four or five smaller projects, none of which will cost over $500 thousand. Each of these smaller, simpler projects will be well under the magic $1 million dollar threshold at which failure becomes a significant factor. The shorter delivery horizon and phasing of the smaller projects also has a massive impact on risk adjustment since the smaller, more immediate change is much more likely to succeed.

SIP introduces a prescriptive architecture to specify how relationships between business functions should project through the technical and data architectures. This is important within the context of capability focused enterprise architecture. In most enterprise architecture approaches, the technical and data architectures end up having little resemblance to the business architecture. In TOGAF, for example, architectural diagrams like that shown in Figure 15 are the norm.

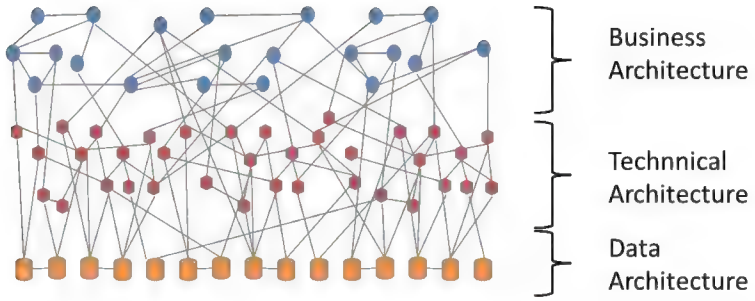

Figure 15. Typical TOGAF architecture.

The typical SIP architecture is quite different. The clean, small, simple partitions that results from analyzing the VRF identified business functions are projected vertically down through the technological architectures. The result is that the technological architectures closely resemble the business architecture above. Because of this strong vertical partitioning process, the typical VRF/SIP architecture resembles Figure 16.

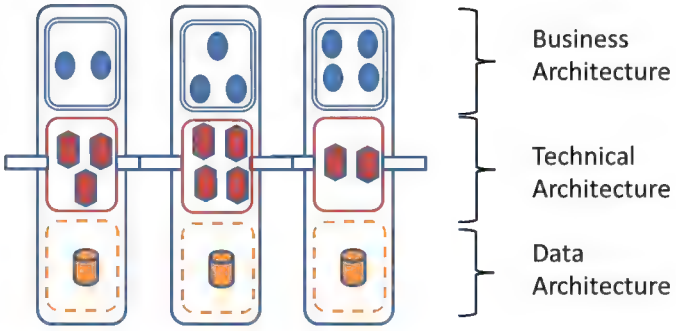

Figure 16. Typical VRF/SIP Architecture.

The VRF/SIP architecture is often described as a Snowman Architecture[9] because each of the autonomous projects is vertically distinct, simple, and autonomous. It takes just a little imagination to see each of the vertical partitions as a snowman. The business dependencies between the projects is reflected at the technical architecture (the torso of the snowman) by asynchronous messages which complete the snowman metaphor by adding arms through which the snowmen interoperate.

It turns out that the Snowman Architecture has a number of important advantages over the traditional non-vertically aligned architecture, such as TOGAF [38]. Besides the lower failure rates that come from limiting the size of each snowman, we have a simpler security model, a more closely aligned business/technology architecture, and more agile systems. As an added

[9] For more information on The Snowman Architecture, see The Snowman Architecture, Part One. [39]

bonus, we have an architecture that is highly efficient to run on a cloud platform which is one of the stated goals of the VRF analysis [41].

SIP completes the capability picture started by VRF[10]. VRF tells us the best way to visualize the enterprise as capabilities and to identify high value projects that can improve the functioning of the capability. SIP tells us the simplest way to break up these projects into a number of vertically partitioned autonomous snowmen that will have a high probability of being delivered successfully.

Once SIP has identified the vertical partitions (or Snowmen), Sally (the SIP specialist) turns the project back over to Vic (the VRF domain expert.) SIP doesn't specify how to track the project, measure the resulting value, or implement the business changes that will be needed. SIP focuses on identifying the best possible business function partitioning and then driving those partitions through the technology architectures. Once the partitioning job is done, VRF takes up the cause once again and focuses on realizing value from the phased roadmap of smaller projects.

Comparison

As you can see, the leading enterprise architecture methodologies are very different in their approaches. Which one is best for your organization? There is no one answer to this question. We'll take you through the 12 criteria that we most often use for comparing and evaluating enterprise architectural methodologies. Not all of these criteria might be relevant to your organization, and some might be more important than others. But, at least, this section can serve as a starting point for your own evaluation.

We'll rank each methodology in each criteria. The ratings will be assigned as follows:

- 1: Does a very poor job in this area
- 2: Does an inadequate job in this area
- 3: Does an acceptable job in this area
- 4: Does a very good job in this area

Keep in mind that these ratings are subjective. Most people will disagree with at least one of our ratings.

Taxonomy completeness refers to how well you can use the methodology to classify the various architectural artifacts. This is almost the entire focus of Zachman. None of the other methodologies focus as much on this area. Ratings:

- Zachman: 4

[10] For more information on the SIP methodology, see The Mathematics of IT Simplification [40]

- TOGAF: 2
- FEA: 2
- VRF/SIP: 1

Process completeness refers to how fully the methodology guides you through a step-by-step process for creating an enterprise architecture. This is almost the entire focus of TOGAF, with its architectural design methodology (ADM). VRF/SIP also focus on this topic. At this point, it seems FEA is de-emphasizing this area. Ratings:

- Zachman: 1
- TOGAF: 4
- FEA: 2
- VRF/SIP: 3

Reference model guidance refers to how useful the methodology is in helping you build a relevant set of reference models. This is almost the entire focus of FEA. TOGAF also provides support; however, we are less impressed with the TOGAF reference models. Ratings:

- Zachman: 1
- TOGAF: 3
- FEA: 4
- VRF/SIP: 1

Practice guidance refers to how much the methodology helps you assimilate the mindset of enterprise architecture into your organization and develop a culture in which it is valued and used. VRF/SIP standout here because of their strong emphasis on cross disciplinary buy-in. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 2
- VRF/SIP: 4

Maturity model refers to how much guidance the methodology gives you in assessing the effectiveness and maturity of different organizations within your enterprise in using enterprise architecture. Ratings:

- Zachman: 1
- TOGAF: 1
- FEA: 3
- VRF/SIP: 2

Business focus refers to whether the methodology will focus on using technology to drive business value, in which business value is specifically defined as either reduced expenses and/or increased income. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 1
- VRF/SIP: 4

Governance guidance refers to how much help the methodology will be in understanding and creating an effective governance model for enterprise architecture. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 3
- VRF/SIP: 2

Partitioning guidance refers to how well the methodology will guide you into effective autonomous partitions of the enterprise, which is an important approach to managing complexity. VRF/SIP stand out here because of SIP's grounding in partitioning theory. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 2
- VRF/SIP: 4

Prescriptive catalog refers to how well the methodology guides you in setting up a catalogue of architectural assets that can be reused in future activities. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 4
- VRF/SIP: 2

Vendor neutrality refers to how likely you are to be dependent on a specific consulting organization when adopting this methodology. A high rating here indicates low vendor lock in. TOGAF stands out here because it is an open standard supported by many vendors. Ratings:

- Zachman: 2
- TOGAF: 4
- FEA: 3

- VRF/SIP: 1

Information availability refers to the amount and quality of free or inexpensive information about this methodology. In this area, VRF and SIP are split. There is considerable information available about SIP but much less about VRF, at this point in time. Ratings:

- Zachman: 2
- TOGAF: 4
- FEA: 2
- VRF/SIP: 3

Time to value refers to the length of time you will likely be using this methodology before you start using it to build solutions that deliver high business value. Ratings:

- Zachman: 1
- TOGAF: 2
- FEA: 1
- VRF/SIP: 4

The criteria and ratings are summarized in Figure 18.

	Ratings			
Criteria	Zachman	TOGAF	FEA	VRF/SIP
Taxonomy Completeness	4	2	2	1
Process Completeness	1	4	2	3
Reference Model Guidance	1	3	4	1
Practice Guidance	1	2	2	4
Maturity Model	1	1	3	2
Business Focus	1	2	1	4
Governance Guidance	1	2	3	2
Partitioning Guidance	1	2	4	4
Prescriptive Catalog	1	2	4	2
Vendor Neutrality	2	4	3	1
Information Availability	2	4	2	3
Time to Value	1	2	1	4

Figure 10. Criteria and Ratings for Each Methodology

One of the important points of Figure 10 is that none of the enterprise architecture methodologies are really complete. Each has strengths and weaknesses.

How do you choose which methodology is best for you? Here is our recommended approach:

1. Go through the rows (criteria) in Figure 10 eliminating any that you feel are not important to your organization.
2. Add any additional rows (criteria) that you feel are important and rate each of the methodologies in that area.
3. Change any of our ratings with which you disagree.

At the end of this exercise you should have a good idea about the strengths and weaknesses of each methodology with respect to your enterprise's needs. If a clear winner emerges, then count yourself lucky. Find a consultant who specializes in helping enterprises implement that methodology and go for it.

For many organizations, there will be no clear winner. For these organizations, we recommend you use a blended approach, in which you create your own enterprise architectural methodology consisting of bits and pieces of each of the methodologies that provide the highest value in your specific areas of concern.

However, you will want a different kind of consultant — one who has a broad perspective on all of these methodologies and specializes in helping enterprises create a methodology that works best given the specific needs and political realities of that enterprise.

Conclusion

This paper has covered a broad introduction to the field of enterprise architecture. The history of the field goes back twenty-five years, but the field is still evolving — and rapidly so. Two of the four major approaches have undergone major changes in just the last two years (VRF/SIP and FEA).

As we have shown, these methodologies are quite different from each other, both in goals and in approach. This is good news and bad.

It is bad news in that it increases the difficulty for many organizations in choosing one single enterprise architectural methodology. How do you choose between methodologies that have so little in common? Choosing between Zachman and TOGAF, for example, is like choosing between spinach and hammers.

But the good news is that these methodologies can be seen as complementing each other. For many organizations, the best choice is some or all of these methodologies, blended together in a way that works well within that organization's constraints. This white paper should provide a good starting place for understanding the value of each of these methodologies and how they can complement each other.

Whichever route you choose, remember that enterprise architecture is a path, not a destination. An enterprise architecture has no value unless it delivers real business value as quickly as possible. One of the most important goals of any enterprise architecture is to bring the business side and the technology sides together so that both are working effectively toward the same goals.

In many organizations, there is a culture of distrust between the technology and business folks. No enterprise-architecture methodology can bridge this divide unless there is a genuine commitment to change. *That commitment must come from the highest level of the organization.* Methodologies cannot solve people-problems, they can only provide a framework within which those problems can be solved.

But once you have that commitment to change, an enterprise architecture methodology can be a valuable tool for guiding that change. This change can manifest itself in many ways. Some of the predicted benefits from a successfully implemented enterprise architectural include:

- Improvements in using IT to drive business adaptability
- Closer partnership between business and IT groups
- Improved focus on organizational goals
- Improved morale as more individuals see a direct correlation between their work and the organization's success
- Reduced numbers of failed IT systems
- Reduced complexity of existing IT systems
- Improved agility of new IT systems
- Closer alignment between IT deliverables and business requirements

It is obvious that an organization that does well in these key areas will be more successful than one that doesn't. This is true regardless of whether success is measured with tangibles, such as profitability and return on investment, or intangibles, such as customer satisfaction and employee turnover.

The starting point for any enterprise architecture is some critical self-analysis. Does your organization spend too much money building IT systems that deliver inadequate business value? Is IT seen as improving business agility or hampering it? Is there a growing divide between your business and IT folks? And then, perhaps the most important question of all: is your organization truly committed to solving these problems and does that commitment come from the highest levels of the organization. If the answer to all of these questions is "yes", then enterprise architecture is your path. It's up to you to take that next step.

Glossary

- **ADM (Architectural Development Method)** – A process for creating an enterprise architecture that is part of the TOGAF standard.
- **application architecture** – The architecture of a specific application.
- **architect** – One whose responsibility is the design of an architecture and the creation of an architectural description.
- **architecture** – The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution (from IEEE-1471-2000).
- **architectural artifact** – A specific document, report, analysis, model, or other tangible that contributes to an architectural description.
- **architectural description** – A collection of products (artifacts) to document an architecture.
- **architectural taxonomy** – A methodology for organizing and categorizing architectural artifacts.
- **architectural framework** – A skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other, and provides generic definitions for what those artifacts might look like.
- **architectural process** – A defined series of actions directed to the goal of producing either an architecture or an architectural description.
- **architectural methodology** – A generic term that can describe any structured approach to solving some or all of the problems related to architecture.
- **business architecture** – An architecture that deals specifically with business processes and business flow.
- **business reference model (BRM)** – A FEA term that gives a business view of the various functions of the federal government.
- **business services segment** – A FEA term that refers to a *segment* that is foundational to most, if not all, political organizations, such as financial management.
- **capability** – A collection of business functionality that works together to deliver some specific business value.
- **capability centric approach** – An approach to enterprise architecture that focuses on analyzing the capabilities of an organization with the goal of maximizing the value that each returns.
- **CIO** — Chief Information Officer, the executive in charge of information technology in a corporation.
- **CIO Council** – A council consisting of CIO's from each of the federal governmental agencies that coordinates work related to common interests.
- **Clinger/Cohen Act** — See *Information Technology Management Reform Act.*
- **common object request broker architecture (CORBA)** – A component-oriented system designed and evangelized by the OMG.
- **common systems architectures** – A TOGAF term referring to an architecture that are common to many, but not all types of enterprises. In contrast to *foundation architectures* and *industry architectures*.
- **component reference model (CRM)** – A FEA term that gives an IT view of systems that support business functionality.

- **data architecture** – The architecture of the data (typically stored in databases) owned by the enterprise.
- **DCOM -** A component-oriented system designed and evangelized by Microsoft.
- **enterprise architect** – An architect who is who specializes in enterprise architectures.
- **enterprise architecture** — An architecture in which the system in question is the whole enterprise, especially the business processes, technologies, and information systems of the enterprise.
- **enterprise service** – A FEA term referring to a well-defined function that spans political boundaries, such as security management.
- **FEA** – See *Federal Enterprise Architecture*.
- **FEAF** — See *Federal Enterprise Architectural Framework*.
- **FEAPMO** – The organization within the OMB that owns and administers the Federal Enterprise Architecture.
- **Federal Enterprise Architecture** – An architectural description of the enterprise architecture of the U.S. federal government that includes various reference models, processes for creating organizational architectures that fit in with the federal enterprise architecture, and a methodology for measuring the success of an organization in using enterprise architectures.
- **Federal Enterprise Architectural Framework (FEAF)** — An enterprise architectural framework used by the United States federal government to describe how the various governmental agencies and their IT systems are related to each other.
- **Federal Architecture Program EA Assessment Framework** – A benchmark used by the OMB to measure the effectiveness of governmental bodies in using enterprise architecture.
- **foundation architecture** – A term used by TOGAF to refer to the most generic of architectures that can be used by any IT organization. In contrast to *common systems architectures*.
- **GAO** — See *General Accountability Office*.
- **Gartner** – An IT research and advisory organization.
- **gateway** — A transfer point of an autonomous system from which messages from the outside world are received or through which messages to the outside world are sent.
- **General Accountability Office** — A branch of the United States Government that is responsible for monitoring the effectiveness of different organizations within the U.S. Government.
- **industry architecture** – A TOGAF term that refers to a architecture that is common to most enterprises within an industry. In contrast to a *common systems architecture* and an *organizational architecture*.
- **Information Technology Management Reform Act** — An act passed by the United States Congress in 1996 that requires all governmental organizations to use effective strategies and frameworks for developing and maintaining IT resources.
- **OMB** (Office of Management and Budget) – Part of the Executive Office of the President of the United States which serves the function of presidential oversight on federal agencies.
- **organizational architecture** – A TOGAF term that applies to an architecture that is specific to a particular organization. In contrast to an *industry architecture*.
- **performance reference model (PRM)** – A FEA term that gives standard ways of describing terms related to measuring value.
- **perspective centric approach** – An approach to enterprise architecture that focuses on aligning different organizational perspectives.

- **process centric approach** – An approach to enterprise architecture that focuses on creating a well defined process for modeling the enterprise.
- **Return on Investment** — A measure (in percent) of the business value of a project based on the increase in profit (either because of increased income or decreased expenses) divided by the cost of the project. For example, a project with a cost of $100,000 that returned $200,000 in increased profit has an ROI of 200%
- **ROI** — see *Return on Investment*.
- **segment** – A FEA term that refers to a major line-of-business functionality, such as human resources, that might be shared across organizations.
- **standardization centric approach** – An approach to enterprise architecture that focuses on defining standard ways of describing different processes, reference models, and common services.
- **standards information base (SIB)** – A TOGAF term which refers to a collection of information about standards, particularly in the area of open source.
- **TAFIM (Technical Architecture Framework for Information Management)** — An architectural framework developed by the Department of Defense and officially discontinued in 2000.
- **technical architecture** – Usually refers to the architecture of the technical infrastructure within which applications run and interact.
- **technical reference model (TRM)** – Part of TOGAF, a reference model that gives a common language for various pieces of IT architecture. This term is also used for a similar meaning within *FEA*.
- The Open Group Architectural Framework – see *TOGAF*.
- **TOGAF (The Open Group Architectural Framework) 8.1** — An architectural methodology that is controlled by The Open Group.
- **Zachman Framework for Enterprise Architectures™** — An architectural framework in which an enterprise is modeled as thirty or thirty-six cells, each of which represents an intersection between a stakeholder perspective and an abstraction.

References

01 – IEEE Standard 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.

02 – A framework for information systems architecture, by J.A. Zachman in IBM Systems Journal, 26 3, 1987.

03 – U.S. Department Of Defense. *Technical Architecture Framework For Information Management (TAFIM) Volumes 1-8*, Version 2.0. Reston, VA: DISA Center for Architecture, 1994.

04 – 1996 Clinger Cohen Act (PL 107-347) see http://thomas.loc.gov/

05 – *Federal Enterprise Architecture Framework Version 1.1*, September 1999 by The Chief Information Officers Council A04

06 – 25 Point Implementation Plan to Reform Federal Information Technology Management by Vivek Kundra available at http://www.dhs.gov/sites/default/files/publications/digital-strategy/25-point-implementation-plan-to-reform-federal-it.pdf.

07 – Cloud Optimized Architectures for the Public Sector (CUEC Position Paper, June 2011) by Roger Sessions available at http://www.objectwatch.com/white_papers.htm#cloud.

08 – The Common Approach to Federal Enterprise Architecture (May 2012) available at http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/common_approach_to_federal_ea.pdf.

09 – INFORMATION TECHNOLOGY The Federal Enterprise Architecture and Agencies' Enterprise Architectures Are Still Maturing, GAO Testimony Before the Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, Committee on Government Reform, House of Representatives, Might 19, 2004.

10 – Information Technology: FBI Is Taking Steps to Develop an Enterprise Architecture, but Much Remains to Be Accomplished GAO-05-363 September 9, 2005

11 – DOD Business Systems Modernization: Long-Standing Weaknesses in Enterprise Architecture Development Need to Be Addressed GAO-05-702 July 22, 2005

12 – Homeland Security: Progress Continues, but Challenges Remain on Department's Management of Information Technology, Testimony Before Congressional Subcommittees, Randolph C. Hite, Director, Information Technology Architecture and Systems Issues, GAO, March 29, 2006

13 – Business Modernization: Some Progress Made toward Implementing GAO Recommendations Related to NASA's Integrated Financial Management Program GAO-05-799R September 9, 2005

14 – *Software Fortresses; Modeling Enterprise Architectures* by Roger Sessions published by Microsoft Press 2003.

15 – The Next Revolution in Productivity by Ric Merrifield, Jack Calhoun, and Dennis Stevens in The Harvard Business Review June 2008.

16 – "framework." The American Heritage® Dictionary of the English Language, Fourth Edition. Houghton Mifflin Company

17 – "taxonomy." *The American Heritage® Dictionary of the English Language*, Fourth Edition. Houghton Mifflin Company.

18 – The Framework for Enterprise Architecture: Background, Description and Utility by John A. Zachman, published by Zachman Institute for Framework Advancement (ZIFA) Document ID 810-231-0531

19 – *Enterprise Architecture Using the Zachman Framework* by Carol O'Rourke, Neal Fishman, and Warren Selkow. Published by Thomson Course Technology 2003. ISBN 0-619-06446-3

20 – Interview with John Zachman by Roger Sessions, Editor-in-Chief, in *Perspectives of the International Association of Software Architects*, Issue 6.

21 – A framework for information systems architecture, by J.A. Zachman in IBM Systems Journal, 26 3, 1987.

22 – Extending and formalizing the framework for information systems architecture, by J.F. Sowa and J.A. Zachman in IBM Systems Journal, 31 3, 1992.

23 – www.opengroup.org.

24 – www.opengroup.org/togaf/

25 – *Guide to Enterprise IT Architecture* by Col Perks and Tony Beveridge, Springer, published 2003, ISBN 0-387-95132-6.

26 – www.opengroup.org/togaf/cert/

27 – *Guide to Enterprise IT Architecture* by Col Perks and Tony Beveridge, Springer, published 2003, ISBN 0-387-95132-6.

28 – TOGAF Version 9.1.

29 – FEA Consolidated Reference Model Document Version 2.1, December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

30 – A Practical Guide to Federal Enterprise Architecture by the CIO Council, Version 1.0, February 2001.

31 – FEA Practice Guidance, December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

32 – FEA Consolidated Reference Model Document Version 2.1, December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

33 – The Data Reference Model, Version 2.0, November 2005, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

34 – FEA Consolidated Reference Model Document Version 2.1, December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

35 – FEA Practice Guidance, December 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.

36 – Federal Enterprise Architecture Program EA Assessment Framework 2.1, Dec 2006

37 – The Relationship Between IT Project Size and Failure Rates, Presentation by Roger Sessions (January 2012) available at http://simplearchitectures.blogspot.com/2012/01/web-short-relationship-between-it.html

38 – The Snowman Architecture Part Three: The Technical Benefits (blog) by Roger Sessions. Available at http://simplearchitectures.blogspot.com/2012/12/the-snowman-architecture-part-three.html

39 – The Snowman Architecture, Part One: Overview by Roger Sessions. Blog, 2012 available at http://simplearchitectures.blogspot.com/2012/09/snowman-architecture-part-one-overview.html.

40 – The Mathematics of IT Simplification. White paper by Roger Sessions April 2011. Available at http://www.objectwatch.com/white_papers.htm#Math.

41 – See Cloud Optimized Architectures for the Public Sector: A CUEC Position Paper. Position Paper by Roger Sessions (June 2011) Available at http://www.objectwatch.com/white_papers.htm#cloud